Working Paper Series
ISSN 1177-777X

AN ALGORITHM FOR THE SYNTHESIS OF LEAST RESTRICTIVE CONTROLLABLE SUPERVISORS FOR EXTENDED FINITE-STATE MACHINES

Robi Malik and Marcelo Teixeira

Working Paper: 01/2016
January 13, 2016

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, 3240
New Zealand

AN ALGORITHM FOR THE SYNTHESIS OF LEAST RESTRICTIVE CONTROLLABLE SUPERVISORS FOR EXTENDED FINITE-STATE MACHINES

Robi Malik
Department of Computer Science
The University of Waikato
Hamilton, New Zealand
robi@waikato.ac.nz

Marcelo Teixeira
Academic Department of Informatics
Federal University of Technology Paraná
Pato Branco, Brazil
marceloteixeira@utfpr.edu.br

January 13, 2016

Abstract

This working paper proposes an algorithm for the synthesis of modular supervisors using extended finite-state machines, i.e., state machines with variables and guards on the transitions. Synthesis is performed by iteratively selecting components from a synchronous composition until a least restrictive controllable solution is obtained. This method is usually faster and produces smaller supervisors than standard monolithic synthesis, while offering the modelling benefits of variables. An example of manufacturing system control illustrates the approach.

1 Introduction

Supervisory Control Theory [10] provides a framework for the *synthesis*, i.e., automatic computation, of *supervisors* or *controllers* for discrete event systems. The theory has been generalised for *Extended Finite-state Machines* (EFSMs) [3,7], which include *variables* and improve modelling capabilities for systems with data dependency or software. Several synthesis algorithms for EFSMs have been proposed [5, 6, 8]. The straightforward synthesis algorithms explore the full system state space, including all possible combinations of variable values, and this can result in prohibitively many states. This complexity can be avoided to some extent using *symbolic* representation [6] or *abstraction* [11, 13].

This working paper focuses on the synthesis of *least restrictive* and *controllable* supervisors, i.e., considering only prefix-closed behaviours, in a *modular* setting, where the system model consists of several interacting EFSM components. In this case, efficient solutions for

systems without variables are known that restrict the synthesis to subsystems [1, 2, 9]. Once appropriate subsystems have been identified, the supervisors synthesised for the subsystems together achieve the least restrictive controllable behaviour of the entire system.

These solutions do not generalise directly to EFSMs, because the composition with other components may introduce new variable assignments and increase behaviour. Therefore, this working paper proposes the use of *abstractions* that capture the possible variable changes outside of the subsystem considered. This results in an incremental procedure that only works with components needed to ensure a controllable and maximally permissive supervisor.

In the following, Section 2 introduces an improved version of the algorithm [1] for modular synthesis of ordinary finite-state machines, and Section 3 generalises this algorithm for EFSMs. Afterwards, Section 4 illustrates the approach with an example, and Section 5 concludes.

2 Finite-State Machines

2.1 Definitions

A *finite-state machine (FSM)* is a tuple $F = \langle \Sigma, Q, Q^\circ, \to \rangle$, where Σ is a finite set of *events*, Q is a finite set of *states*, $Q^\circ \subseteq Q$ is the set of *initial states*, and $\to \subseteq Q \times \Sigma \times Q$ is the *transition relation*.

The transition relation is written in infix notation, where $x \xrightarrow{\sigma} y$ means the existence of a transition from state $x \in Q$ to $y \in Q$ with event $\sigma \in \Sigma$. This notation is extended to *traces* $s \in \Sigma^*$ in the standard way. Furthermore, given state sets $X, Y \subseteq Q$, the notation $X \xrightarrow{s} Y$ means $x \xrightarrow{s} y$ for some states $x \in X$ and $y \in Y$, and $X \to Y$ means $X \xrightarrow{s} Y$ for some $s \in \Sigma^*$, and $X \xrightarrow{s}$ means $X \xrightarrow{s} Y$ for some Y, and $F \xrightarrow{s} X$ means $Q^\circ \xrightarrow{s} X$. A trace $s \in \Sigma^*$ is *accepted* by the FSM if $F \xrightarrow{s}$, and the *language* or *behaviour* of F is the set of all traces it accepts, $\mathcal{L}(F) = \{\, s \in \Sigma^* \mid F \xrightarrow{s} \,\}$.

In this working paper, FSMs do not have accepting states, and all languages are prefix-closed: trace $s \in \Sigma^*$ is a *prefix* of $t \in \Sigma^*$ if $t = su$ for some $u \in \Sigma^*$, and the language $L \subseteq \Sigma^*$ is *prefix-closed*, if all prefixes of traces $t \in L$ are contained in L.

FSMs executing in parallel are synchronised in lock-step [4]. The *synchronous composition* of two FSMs $F_1 = \langle \Sigma, Q_1, Q_1^\circ, \to_1 \rangle$ and $F_2 = \langle \Sigma, Q_2, Q_2^\circ, \to_2 \rangle$ with the same event set Σ is

$$F_1 \parallel F_2 = \langle \Sigma, Q_1 \times Q_2, Q_1^\circ \times Q_2^\circ, \to \rangle \ , \tag{1}$$

where $(x_1, x_2) \xrightarrow{\sigma} (y_1, y_2)$ if and only if $x_1 \xrightarrow{\sigma} x_2$ and $y_1 \xrightarrow{\sigma} y_2$. FSMs with different event sets can be composed after the *selfloop* operation [14], adding selfloop transitions $x \xrightarrow{\sigma} x$ with the missing events to all states of an FSM; this is not considered in this section for the sake of brevity, and all FSMs are assumed to have the same event set Σ. In this case, it is clear that synchronous composition of FSMs results in the intersection of their languages, $\mathcal{L}(F_1 \parallel F_2) = \mathcal{L}(F_1) \cap \mathcal{L}(F_2)$.

For the purpose of control, the event set is partitioned into the sets Σ_c of *controllable* events and Σ_u of *uncontrollable* events. Controllable events can be disabled by a controlling agent, while uncontrollable events occur spontaneously. A prefix-closed *specification* language $K \subseteq \Sigma^*$ is Σ_u-controllable [10] with respect to a prefix-closed *plant* language $L \subseteq \Sigma^*$ if $K\Sigma_\mathrm{u} \cap L \subseteq K$, i.e., if every uncontrollable event continuation possible in L is also possible in K.

If a language K is not controllable, the task of *synthesis* is to find a controllable sublanguage $K' \subseteq K$. It is a classical result of *supervisory control theory* [10] that the union of controllable languages is again controllable, and there exists a unique *supremal controllable sublanguage* of any given language,

$$\sup\mathcal{C}(L, K, \Sigma_\mathrm{u}) = \bigcup \{\, K' \subseteq L \cap K \mid K' \text{ is } \Sigma_\mathrm{u}\text{-controllable with respect to } L \,\} \,. \tag{2}$$

It is common to require that the result of synthesis is contained in the plant language L, which is enforced by the intersection $L \cap K$ in (2). If the plant and specification are given by FSMs G and E, a standard algorithm [10] with time complexity polynomial in the number of transitions of $G \parallel E$ can construct an FSM that accepts (2). This FSM is denoted $\sup\mathcal{C}(G, E, \Sigma_\mathrm{u})$ in the following. It can be used as a so-called *supervisor*, which restricts the plant through synchronous composition, enforcing the specification by disabling only controllable events in the least restrictive way possible.

2.2 Modular Synthesis Algorithm

In the following, it is assumed that the plant is given by several FSMs, $G = G_1 \parallel \cdots \parallel G_n$, and the specification is given by a single FSM E. In this case, the complexity to compute $\sup\mathcal{C}(G, E, \Sigma_\mathrm{u})$ is exponential in the number n of plant components due to the exponential number of states in the synchronous composition.

It has been proposed [1,2] to mitigate this complexity by identifying an appropriate subset of the plants to perform synthesis with. Algorithm 1 shows such an approach. It is the basis for the extended finite-state machine synthesis algorithm in the following section.

Here and in the following, for a set $\mathcal{G}$ of state machines, $\Big\|(\mathcal{G})$ denotes the synchronous composition of all elements of $\mathcal{G}$, and $\Big\|(\emptyset) = \langle \Sigma, \{x^\circ\}, \{x^\circ\}, \{x^\circ\} \times \Sigma \times \{x^\circ\} \rangle$ is the neutral element of synchronous composition, a state machine that accepts all events without state change.

The idea of Algorithm 1 is to gradually increase the set of plants and uncontrollable events considered in synthesis. At the beginning, the loop entry condition in line 5 checks whether the specification $S^0 = E$ is controllable by itself. This may succeed if, for example, E has only controllable events, in which case E is returned as the least restrictive solution. Otherwise the loop is entered and performs synthesis with respect to selected subsets $\mathcal{G}^{i+1}$ of plants and Σ_u^{i+1} of uncontrollable events (line 8). When inside the loop, line 5 ensures that the

Algorithm 1: Modular FSM synthesis.

Input: plants $\mathcal{G} = \{G_1, \ldots, G_m\}$; specification E; uncontrollable events Σ_u;

1 $\mathcal{G}^0 \leftarrow \emptyset$;
2 $S^0 \leftarrow E$;
3 $\Sigma_\mathrm{u}^0 \leftarrow \emptyset$;
4 $i \leftarrow 0$;
5 **while** $\mathcal{L}(S^i)$ *is not* Σ_u*-controllable with respect to* $\mathcal{L}(\|(\mathcal{G}^i))$ **do**
6 $\quad \Sigma_\mathrm{u}^{i+1} \leftarrow \Sigma_\mathrm{u}^i \cup \{\, \mu \in \Sigma_\mathrm{u} \mid \|(\mathcal{G}^i) \parallel S^i \to (x_G, x_S) \text{ and } x_G \xrightarrow{\mu} \text{ and } x_S \not\xrightarrow{\mu} \,\}$;
7 $\quad \mathcal{G}^{i+1} \leftarrow \{\, G' \in \mathcal{G} \mid G' \to x_G \not\xrightarrow{\mu} \text{ for some } \mu \in \Sigma_\mathrm{u}^{i+1} \,\}$;
8 $\quad S^{i+1} \leftarrow \mathrm{sup}\mathcal{C}(\|(\mathcal{G}^{i+1}), E, \Sigma_\mathrm{u}^{i+1})$;
9 $\quad i \leftarrow i + 1$;
10 **end**
11 **return** S^i

current result S^i is not controllable with respect to the full set Σ_u of uncontrollable events. Thus, S^i disables some event $\mu \in \Sigma_\mathrm{u} \setminus \Sigma_\mathrm{u}^{i+1}$, which really is uncontrollable but was assumed controllable in synthesis. These events are treated as uncontrollable in the next iteration (line 6). To ensure the least restrictive result, all plants that in some state may disable one of these uncontrollable events are also included (line 7).

The procedure continues until a Σ_u-controllable solution is found. Termination is guaranteed, because the set Σ_u^i of included uncontrollable events increases with every iteration. As the result is Σ_u-controllable with respect to a subset of plants, it is also controllable with respect to the full plant, as stated in the following lemma.

Lemma 1 Let $K, L_1, L_2 \subseteq \Sigma^*$ be prefix-closed languages, and let $\Sigma_\mathrm{u} \subseteq \Sigma$ be a set of events. If K is Σ_u-controllable with respect to L_1, then K is Σ_u-controllable with respect to $L_1 \cap L_2$.

Proof. See Proposition 3 in [2]. □

The approach [1] ensures least restrictiveness by including all plants that share an uncontrollable event with the specification, or with a plant already included. Algorithm 1 improves on this in line 6, by considering only uncontrollable events that cause a controllability problem, and in line 7, by only adding plants that disable an uncontrollable event, as opposed to plants that have it in their event set. This is enough to ensure that supervisor S^i remains an over-approximation of the least restrictive synthesis result after each iteration i.

Lemma 2 Before and after each iteration of Algorithm 1, it holds that

$$\mathcal{L}(\|(\mathcal{G}) \parallel S^i) \supseteq \mathrm{sup}\mathcal{C}(\mathcal{L}(\|(\mathcal{G})), \mathcal{L}(E), \Sigma_\mathrm{u}) \ . \tag{3}$$

Proof. Before the first iteration, $i = 0$, it holds according to line 1 of Algorithm 1 that $\mathcal{L}(\|(\mathcal{G}) \parallel S^0) = \mathcal{L}(\|(\mathcal{G}) \parallel E) \supseteq \mathcal{L}(\sup\mathcal{C}(\mathcal{L}(\|(\mathcal{G})), \mathcal{L}(E), \Sigma_u))$.

Otherwise, it holds after line 8 that $S^{i+1} = \sup\mathcal{C}(\|(\mathcal{G}^{i+1}), E, \Sigma_u^{i+1})$. This firstly implies $\sup\mathcal{C}(\mathcal{L}(\|(\mathcal{G})), \mathcal{L}(E), \Sigma_u) \subseteq \mathcal{L}(\|(\mathcal{G})) \cap \mathcal{L}(E) \subseteq \mathcal{L}(\|(\mathcal{G}^{i+1})) \cap \mathcal{L}(E)$, and secondly it follows that $\sup\mathcal{C}(\mathcal{L}(\|(\mathcal{G})), \mathcal{L}(E), \Sigma_u)$ is Σ_u^{i+1}-controllable with respect to $\|(\mathcal{G}^{i+1})$. To see the latter, let $s \in \sup\mathcal{C}(\mathcal{L}(\|(\mathcal{G})), \mathcal{L}(E), \Sigma_u)$ and $\mu \in \Sigma_u^{i+1}$ such that $s\mu \in \mathcal{L}(\|(\mathcal{G}^{i+1}))$. Then $s \in \sup\mathcal{C}(\mathcal{L}(\|(\mathcal{G})), \mathcal{L}(E), \Sigma_u) \subseteq \mathcal{L}(\|(\mathcal{G}))$. As $\mu \in \Sigma_u^{i+1}$ it follows from line 7 that μ is enabled in all accessible states of all plants outside of $\mathcal{G}^{i+1}$, i.e., not only $s\mu \in \mathcal{L}(\|(\mathcal{G}^{i+1}))$ but also $s\mu \in \mathcal{L}(\|(\mathcal{G}))$. Therefore, $s\mu \in \sup\mathcal{C}(\mathcal{L}(\|(\mathcal{G})), \mathcal{L}(E), \Sigma_u)$ as $\sup\mathcal{C}(\mathcal{L}(\|(\mathcal{G})), \mathcal{L}(E), \Sigma_u)$ is Σ_u-controllable with respect to $\|(\mathcal{G})$. It follows that $\sup\mathcal{C}(\mathcal{L}(\|(\mathcal{G})), \mathcal{L}(E), \Sigma_u) \subseteq \sup\mathcal{C}(\|(\mathcal{G}^{i+1}), E, \Sigma_u^{i+1}) = \mathcal{L}(S^{i+1})$, and then also $\mathcal{L}(\|(\mathcal{G}) \| S^{i+1}) \supseteq \mathcal{L}(\|(\mathcal{G})) \cap \sup\mathcal{C}(\mathcal{L}(\|(\mathcal{G})), \mathcal{L}(E), \Sigma_u) = \sup\mathcal{C}(\mathcal{L}(\|(\mathcal{G})), \mathcal{L}(E), \Sigma_u)$. □

Lemmas1 and 2 are combined to obtain Proposition 3, which confirms that Algorithm 1 returns an FSM implementing the least restrictive supervisor. As synthesis within the loop only includes a part of the plant, the remaining plants have to be composed with the result to get the exact supremal controllable sublanguage.

Proposition 3 Upon termination of Algorithm 1, it holds that

$$\mathcal{L}(\|(\mathcal{G}) \parallel S^i) = \sup\mathcal{C}(\mathcal{L}(\|(\mathcal{G})), \mathcal{L}(E), \Sigma_u) \ . \tag{4}$$

Proof. By the loop-entry condition (line 5 of Algorithm 1), it is clear that $\mathcal{L}(S^i)$ is Σ_u-controllable with respect to $\mathcal{L}(\mathcal{G}^i)$ upon termination of the loop. By Lemma 1, it follows that $\mathcal{L}(S^i)$ is Σ_u-controllable with respect to $\mathcal{L}(\mathcal{G})$, and then $\mathcal{L}(\|(\mathcal{G}) \parallel S^i)$ is also Σ_u-controllable with respect to $\mathcal{L}(\mathcal{G})$ [2]. As furthermore $\mathcal{L}(S^i) \subseteq \mathcal{L}(E)$ from lines 2 and 8, it holds that $\mathcal{L}(\|(\mathcal{G}) \parallel S^i) \subseteq \mathcal{L}(\|(\mathcal{G}) \parallel E)$, and therefore $\mathcal{L}(\|(\mathcal{G}) \parallel S^i) \subseteq \sup\mathcal{C}(\mathcal{L}(\|(\mathcal{G}), \mathcal{L}(E), \Sigma_u)$. The converse inclusion has been shown in Lemma 2, hence the equality (4) holds. □

3 Extended Finite-State Machines

Extended finite-state machines (EFSMs) add to FSMs *variables* and the ability to read and update these variables on the occurrence of transitions [3,7]. Examples of such state machines are shown in Figure 4 on page 26.

3.1 Variables and Updates

An *update* is a formula constructed from variables, integer constants, Boolean literals, and the usual arithmetic and logic connectives. The set of all update formulas is denoted by Π.

A *variable* v is an entity associated with a finite discrete *domain* $\mathrm{dom}(v)$ and an initial value $\hat{v}^\circ \in \mathrm{dom}(v)$. Let $V = \{v_0, \ldots, v_n\}$ be the set of variables with domain $\mathrm{dom}(V) =$

$\mathrm{dom}(v_0) \times \cdots \times \mathrm{dom}(v_n)$. An element $\hat{v}$ of $\mathrm{dom}(V)$ is also considered as a *valuation* that assigns to each variable $v \in V$ a value $\hat{v}(v) \in \mathrm{dom}(v)$, and by extension a truth value to each update. The *initial valuation* is $\hat{v}^\circ \in \mathrm{dom}(V)$ with $\hat{v}^\circ(v) = \hat{v}^\circ$ for each $v \in V$. An update is *satisfiable* if it is true for at least one valuation, otherwise *unsatisfiable*, and *valid* if it is true for all valuations of its variables. The *restriction* of a valuation $\hat{v} \in \mathrm{dom}(V)$ to $W \subseteq V$ is $\hat{v}_{|W} \in \mathrm{dom}(W)$ with $\hat{v}_{|W}(v) = \hat{v}(v)$ for all $v \in W$. Two valuations $\hat{v} \in \mathrm{dom}(V)$ and $\hat{w} \in \mathrm{dom}(W)$ can be combined to give $\hat{v} \oplus \hat{w} \in \mathrm{dom}(V \cup W)$ where $(\hat{v} \oplus \hat{w})(v) = \hat{v}(v)$ for $v \in V$ and $(\hat{v} \oplus \hat{w})(w) = \hat{w}(w)$ for $w \in W \setminus V$.

A second set of variables, called *next-state* variables and denoted $V' = \{\, v' \mid v \in V \,\}$ is used to describe the values of the variables after a transition. Variables in V are also referred to as *current-state* variables to differentiate them from the next-state variables in V'. The next-state variable v' has the same domain as its current-state variable v. Given $\hat{v} \in \mathrm{dom}(V)$, the valuation $\hat{v}' \in \mathrm{dom}(V')$ is defined by $\hat{v}'(v') = \hat{v}(v)$ for all $v \in V$. For an update $p \in \Pi$, the term $\mathrm{vars}(p)$ denotes the set of all variables that occur as current-state or next-state variable in p, and $\mathrm{vars}'(p)$ denotes the set of all variables whose corresponding next-state variables occur in p. For example, if p is the update $x' = y+1$, then $\mathrm{vars}(p) = \{x, y\}$ and $\mathrm{vars}'(p) = \{x\}$.

3.2 EFSM Definition and Operations

Definition 1 An *Extended finite-state machine (EFSM)* is a tuple $F = \langle \Sigma, Q, Q^\circ, \to \rangle$, where Σ is a finite set of events, Q is a finite set of *locations*, $Q^\circ \subseteq Q$ is the set of *initial locations*, and $\to \subseteq Q \times \Sigma \times \Pi \times Q$ is the *extended transition relation*.

A transition between locations $x, y \in Q$ with event $\sigma \in \Sigma$ and update $p \in \Pi$ is written $x \overset{\sigma:p}{\to} y$. It can occur if F is in location x and the update p evaluates to true, and when it occurs, F changes its location to y while updating the variables in $\mathrm{vars}'(p)$ in accordance with p; variables not in $\mathrm{vars}'(p)$ remain unchanged.

For example, let x be a variable with domain $\mathrm{dom}(x) = \{0, \ldots, 5\}$. A transition with update $x' = x + 1$ changes the variable x by adding 1 to its current value, if it currently is less than 5. Otherwise (if $x = 5$) the transition is disabled. The update $x = 3$ disables a transition unless $x = 3$ in the current state, and the value of x in the next state is unchanged. Differently, the update $x' = 3$ always enables its transition, and the value of x in the next state is forced to be 3.

Given an EFSM F and event $\sigma \in \Sigma$, the *referenced variable set* is $\mathrm{vars}(F, \sigma) = \bigcup\{\, \mathrm{vars}(p) \mid x \overset{\sigma:p}{\to} y \,\}$, and $\mathrm{vars}(\mathcal{F}, \sigma) = \bigcup_{F' \in \mathcal{F}} \mathrm{vars}(F', \sigma)$ for a set $\mathcal{F}$ of EFSMs. Furthermore, $\mathrm{vars}(F) = \bigcup_{\sigma \in \Sigma} \mathrm{vars}(F, \sigma)$, and analogous notation is defined for vars'.

This working paper imposes some syntactic restrictions on system models, which are needed for the modularity results.

Definition 2 Let $F = \langle \Sigma, Q, Q^\circ, \to \rangle$ be an EFSM.

- F is *normalised*, if for any two transitions $x_1 \xrightarrow{\sigma:p_1} y_1$ and $x_2 \xrightarrow{\sigma:p_2} y_2$ with the same event $\sigma \in \Sigma$, it holds that $\mathrm{vars}'(p_1) = \mathrm{vars}'(p_2)$.
- F is *pure* if $\mathrm{vars}'(F) = \emptyset$.
- F is *state-deterministic* if $|Q^\circ| \leq 1$, and for all transitions $x \xrightarrow{\sigma:p_1} y_1$ and $x \xrightarrow{\sigma:p_2} y_2$ such that $p_1 \wedge p_2$ is satisfiable, it holds that $y_1 = y_2$.

In a normalised EFSM, the set of variables changed by an event is the same on all transitions. This assumption helps to maintain the implicitly unchanged variables after abstraction and synchronous composition. Every EFSM can be transformed into a normalised EFSM by a process of renaming similar to normalisation [7]. As a stronger condition, a pure EFSM cannot assign any variables, it only restricts events. State-determinism is needed for supervisors to track the location of the plant by the observation of events and variable values. It ensures that the target locations are uniquely determined from the source location, event, and variable assignment.

In the following, plants are modelled by normalised state-deterministic EFSMs, while specifications are pure state-deterministic EFSMs. The synthesised supervisor is not subject to these requirements and in particular can restrict variable assignments.

An EFSM $F = \langle \Sigma, Q, Q^\circ, \to \rangle$ can be *unfolded* [7, 13] and interpreted as an FSM with state set $Q \times \mathrm{dom}(\mathrm{vars}(F))$. The states $(x, \hat{v})$ consist of a location $x \in Q$ and a valuation $\hat{v} \in \mathrm{dom}(\mathrm{vars}(F))$. A transition between two states, written $(x, \hat{v}) \xrightarrow{\sigma} (y, \hat{w})$, exists if F contains a transition $x \xrightarrow{\sigma:p} y$ such that that the update p is true if the current-state variables are interpreted according to $\hat{v}$ and the next-state variables according to $\hat{w}$, i.e., $(\hat{v} \oplus \hat{w}')(p)$ is true, and all variables that do not appear as next-state variables in the update are unchanged, i.e., $\hat{v}(v) = \hat{w}(v)$ for $v \notin \mathrm{vars}'(p)$. This transition relation is also defined for variables not in $\mathrm{vars}(F)$, which remain unchanged, and for events not in Σ, which are always enabled without changing the EFSM's location or any variables. That is, $(x, \hat{v}) \xrightarrow{\sigma} (x, \hat{v})$, for all $x \in Q$ and $\sigma \notin \Sigma$. The $\to$ notation is extended to traces, state sets, and state machines in the same way as for FSMs. Based on this, the set of *accessible states* of an EFSM F is

$$Q^{\mathrm{acc}}(F) = \{\, (x, \hat{v}) \in Q \times \mathrm{dom}(\mathrm{vars}(F)) \mid F \to (x, \hat{v}) \,\} \,. \tag{5}$$

With the following definition, an EFSM can be *restricted* to a set of unfolded states, symbolically, by rewriting updates to impose new constraints on the variables.

Definition 3 Let $F = \langle \Sigma, Q, Q^\circ, \to \rangle$ be an EFSM, and let $X \subseteq Q \times \mathrm{dom}(\mathrm{vars}(F))$. The *symbolic restriction* of F to X is the EFSM $F \restriction X = \langle \Sigma, Q_{|X}, Q^\circ_{|X}, \to_{|X} \rangle$, where

$$Q_{|X} = \{\, x \in Q \mid (x, \hat{v}) \in X \text{ for some } \hat{v} \in \mathrm{dom}(\mathrm{vars}(F)) \,\} \,; \tag{6}$$

$$Q^\circ_{|X} = \{\, x^\circ \in Q^\circ \mid (x^\circ, \hat{v}^\circ) \in X \,\} \,; \tag{7}$$

and $x \xrightarrow{\sigma:p \wedge q_y}_{|X} y$, if $x, y \in Q_{|X}$, and $x \xrightarrow{\sigma:p} y$, and q_y is an update such that $\mathrm{vars}'(q_y) = \mathrm{vars}'(F)$ and $\hat{v}'(q_y)$ is true if and only if $(y, \hat{v}) \in X$.

Definition 4 Let $F = \langle \Sigma, Q, Q^\circ, \rightarrow \rangle$ be an EFSM. The *accessible sub-EFSM* of F is $\mathrm{Acc}(F) = F \restriction Q^{\mathrm{acc}}(F)$.

When comparing EFSMs, variables must be considered in addition to events, so the following notion of behavioural inclusion replaces language inclusion as used for FSMs.

Definition 5 An EFSM F_1 is *behaviourally included* in another EFSM F_2, written $F_1 \subseteq_{\mathrm{v}} F_2$, if for every path

$$(x_0, \hat{v}_0) \xrightarrow{\sigma_1} (x_1, \hat{v}_1) \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_n} (x_n, \hat{v}_n) \tag{8}$$

in F_1, with $\hat{v}_i \in \mathrm{dom}(\mathrm{vars}(F_1))$, there exists a path

$$(y_0, \hat{w}_0) \xrightarrow{\sigma_1} (y_1, \hat{w}_1) \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_n} (y_n, \hat{w}_n) \tag{9}$$

in F_2 such that $\hat{w}_i \in \mathrm{dom}(\mathrm{vars}(F_2))$ and $\hat{v}_{i\,|V_{12}} = \hat{w}_{i\,|V_{12}}$ for $1 \leq i \leq n$, where $V_{12} = \mathrm{vars}(F_1) \cap \mathrm{vars}(F_2)$.

If F_1 is behaviourally included in F_2 then every path in F_1 corresponds to a path in F_2 with the same events and variable assignments. The two EFSMs typically use the same variables, but if not, only the common variables are required to match. It is an immediate consequence of this definition that symbolic restriction (Definition 3) results in a behaviourally included EFSM, i.e., $F \restriction X \subseteq_{\mathrm{v}} F$.

Definition 6 The *synchronous composition* of two EFSMs $F_1 = \langle \Sigma_1, Q_1, Q_1^\circ, \rightarrow_1 \rangle$ and $F_2 = \langle \Sigma_2, Q_2, Q_2^\circ, \rightarrow_2 \rangle$ is

$$F_1 \parallel F_2 = \langle \Sigma_1 \cup \Sigma_2, Q_1 \times Q_2, Q_1^\circ \times Q_2^\circ, \rightarrow \rangle \ , \tag{10}$$

where $(x_1, x_2) \xrightarrow{\sigma:p_1 \wedge p_2} (y_1, y_2)$ if $x_1 \xrightarrow{\sigma:p_1}_1 y_1$ and $x_2 \xrightarrow{\sigma:p_2}_2 y_2$.

This definition captures EFSMs with different event sets through the extended definition of the transition relation above. Updates in synchronous composition are combined by conjunction. They may cancel each other out, e.g., if $x_1 \xrightarrow{\sigma:x'=0}_1 y_1$ in F_1 and $x_2 \xrightarrow{\sigma:x'=1}_2 y_2$ in F_2, then the conjunction $x' = 0 \wedge x' = 1$ is unsatisfiable, or equivalently there is no such transition in $F_1 \parallel F_2$. Synchronous composition can override the assumption of implicitly unchanged variables in an EFSM. If $x_1 \xrightarrow{\sigma:x=0}_1 y_1$ and $x_2 \xrightarrow{\sigma:x'=x+1}_2 y_2$, e.g., then $(x_1, x_2) \xrightarrow{\sigma:x=0 \wedge x'=x+1} (y_1, y_2)$. So the value of x changes from 0 to 1 in $F_1 \parallel F_2$ although implicitly unchanged in F_1.

It is clear that EFSM synchronous composition is associative and commutative, but it is not idempotent as $F \parallel F = F$ does not generally hold for non-deterministic state machines. If F is a state-deterministic EFSM, then $F \parallel F = F$ holds up to isomorphism, after deletion of transitions with unsatisfiable updates and inaccessible locations.

3.3 EFSM Controllability and Synthesis

For the supervisory control of EFSM systems, this working paper assumes that all variables are controlled by the plant. The plant is modelled by a set of normalised EFSMs that represent the possible system behaviour including all possible variable changes. The specification is typically modelled by one or more pure EFSMs, which only restrict the occurrence of events. The supervisor can also restrict variable changes. The following definition of controllability covers both cases.

Definition 7 Let $G = \langle \Sigma_G, Q_G, Q_G^\circ, \rightarrow_G \rangle$ and $E = \langle \Sigma_E, Q_E, Q_E^\circ, \rightarrow_E \rangle$ be two EFSMs, and let Σ_u be a set of events. E is Σ_u-*controllable* with respect to G, if for all valuations $\hat{v}, \hat{w} \in \mathrm{dom}(\mathrm{vars}(G) \cup \mathrm{vars}(E))$, all states $(x_G, x_E, \hat{v}) \in Q^{\mathrm{acc}}(G \parallel E)$, and all transitions $(x_G, \hat{v}) \xrightarrow{\mu} (y_G, \hat{w})$ in G such that $\mu \in \Sigma_E \cap \Sigma_\mathrm{u}$, there exists a location y_E of E such that $(x_G, x_E, \hat{v}) \xrightarrow{\mu} (y_G, y_E, \hat{w})$ in $G \parallel E$.

Σ_u-controllability means that, from any accessible state in the synchronous composition of the plant G and specification E, if an *uncontrollable* event is eligible in the plant, then it is also eligible in the specification. In addition, the specification must allow any assignment to next-state variables prescribed by the plant. The condition $(x_G, x_E, \hat{v}) \xrightarrow{\mu} (y_G, y_E, \hat{w})$ is applied to the synchronous composition $G \parallel E$, so it requires the plant and specification to be able to take the transition together. This allows a pure specification to follow the plant's assignments to next-state variables.

In the case that an uncontrollable event is not mentioned in the plant, $\mu \notin \Sigma_G$, based on the extended definition of the transition relation, the transition is always possible in the plant and does not change variables. In order to be controllable, the specification must always enable μ without changing any of the plant's variables on its occurrence.

Remark 1 Given a plant G and pure specification E, it can be assumed without loss of generality, that $\mathrm{vars}(E) \subseteq \mathrm{vars}(G)$. This is because any variable v that appears only in E and not in G, cannot appear as next-state variable in G or E as $\mathrm{vars}'(E) = \emptyset$. This variable v remains unchanged on all transitions of the synchronous composition $G \parallel E$, so all its occurrences can be replaced by a constant representing its initial value $\hat{v}^\circ$, resulting in an EFSM system with equivalent behaviour.

If a specification is not controllable, *synthesis* is used to find a a *supervisor*, which restricts the plant while satisfying the specification. The supervisor is an EFSM to be composed with the plant in synchronous composition, and unlike the specification, may include next-state variables on its updates. Thus, the supervisor can disable (controllable) events completely or under certain circumstances, and it can remove some of the plant's variable assignments from a controllable transition.

Definition 8 Let G and E be two EFSMs, and let Σ_u be a set of events. A *supremal supervisor* for E with respect to G and Σ_u is an EFSM S such that

(i) $\mathrm{vars}(S) \subseteq \mathrm{vars}(G) \cup \mathrm{vars}(E)$;

(ii) $G \parallel S \subseteq_\mathrm{v} G \parallel E$;

(iii) S is Σ_u-controllable with respect to G;

(iv) For any EFSM S' that satisfies (ii) and (iii), it holds that $G \parallel S' \subseteq_\mathrm{v} G \parallel S$.

Definition 8 characterises the possible synthesis results for a given plant G and specification E. A correct supervisor can only use variables that appear in the plant or specification (i). This syntactic condition is convenient but not essential, as any extra variables can be unfolded into locations to construct an equivalent supervisor. A correct supervisor must satisfy the specification through behavioural inclusion after composition with the plant (ii), and it must be controllable (iii). It also should be *least restrictive* or *supremal*, i.e., any other supervisor that controllably satisfies the specification should have less possible behaviour, again in composition with the plant (iv).

A supervisor satisfying these four conditions can be computed by means of a standard fixpoint iteration on the unfolded state set of $G \parallel E$, using the following operator.

Definition 9 Let $G = \langle \Sigma_G, Q_G, Q_G^\circ, \rightarrow_G \rangle$ and $E = \langle \Sigma_E, Q_E, Q_E^\circ, \rightarrow_E \rangle$ be two EFSMs, let $V = \mathrm{vars}(G) \cup \mathrm{vars}(E)$, and let Σ_u be a set of events. The *extended synthesis step* operator $\Theta_{G,E,\Sigma_\mathrm{u}} \colon 2^{Q_G \times Q_E \times \mathrm{dom}(V)} \to 2^{Q_G \times Q_E \times \mathrm{dom}(V)}$ with respect to G, E, and Σ_u is defined as

$$\begin{aligned}\Theta_{G,E,\Sigma_\mathrm{u}}(X) = \{\, (x_G, x_E, \hat{v}) \in Q_G \times Q_E \times \mathrm{dom}(V) \mid \text{if } (x_G, \hat{v}) \xrightarrow{\mu} (y_G, \hat{w}) \text{ for some } \mu \in \Sigma_\mathrm{u} \\ \text{and } \hat{w} \in \mathrm{dom}(V), \text{ then there exists } y_E \in Q_E \text{ such that } (x_G, x_E, \hat{v}) \xrightarrow{\mu} \\ (y_G, y_E, \hat{w}) \in X \,\} \,.\end{aligned} \tag{11}$$

For a set X of combinations of locations and variable assignments, the operator $\Theta_{G,E,\Sigma_\mathrm{u}}$ removes from X any uncontrollable states, i.e., states where the plant enables some uncontrollable transitions not enabled by the specification, and any states from where the system could uncontrollably reach some combination of location and valuation not contained in X. It is easy to see that the operator $\Theta_{G,E,\Sigma_\mathrm{u}}$ is monotonic and has a greatest fixpoint $\hat{\Theta}_{G,E,\Sigma_\mathrm{u}}$ [12]. In the finite-state case, this fixpoint is calculated as the limit of the iteration

$$X^0 = Q_Q \times Q_E \times \mathrm{dom}(V) \,; \tag{12}$$

$$X^{j+1} = \Theta_{G,E,\Sigma_\mathrm{u}}(X^j) \,. \tag{13}$$

The result of state-based EFSM synthesis is then obtained by restricting the system to this fixpoint.

Definition 10 Let G and E be two EFSMs, and let Σ_u be a set of events. The *supremal* Σ_u*-controllable sub-EFSM* of G and E is

$$\sup\mathcal{C}(G, E, \Sigma_\mathrm{u}) = \mathrm{Acc}((G \parallel E) \restriction \hat{\Theta}_{G,E,\Sigma_\mathrm{u}}) \ , \tag{14}$$

where $\hat{\Theta}_{G,E,\Sigma_\mathrm{u}}$ is the greatest fixpoint of the operator $\Theta_{G,E,\Sigma_\mathrm{u}}$ from Definition 11.

The following Proposition 4 confirms that this fixpoint indeed gives a correct synthesis result according to Definition 8.

Proposition 4 Let G and E be state-deterministic EFSMs, let E be pure, and let Σ_u be a set of events. Then $\sup\mathcal{C}(G, E, \Sigma_\mathrm{u})$ is a supremal supervisor for E with respect to G and Σ_u.

Proof. Write $S = \sup\mathcal{C}(G, E, \Sigma_\mathrm{u})$. It is to be shown that S satisfies conditions (i)–(iv) in Definition 8.

(i) It follows from Definitions 3 and 10 that $\mathrm{vars}(S) = \mathrm{vars}(\sup\mathcal{C}(G, E, \Sigma_\mathrm{u})) \subseteq \mathrm{vars}(G) \cup \mathrm{vars}(E)$.

(ii) Note that $\mathrm{vars}(G \parallel S) = \mathrm{vars}(G) \cup \mathrm{vars}(G) \cup \mathrm{vars}(E) = \mathrm{vars}(G \parallel E)$. Let

$$(x_G^0, (y_G^0, x_E^0), \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, (y_G^n, x_E^n), \hat{v}^n) \tag{15}$$

be a path in $G \parallel S = G \parallel \sup\mathcal{C}(G, E, \Sigma_\mathrm{u})$. Then $x_G^k = y_G^k$ for $0 \leq k \leq n$ by the state-determinism of G. It follows that

$$(x_G^0, x_E^0, \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, x_E^n, \hat{v}^n) \tag{16}$$

is a path in $\sup\mathcal{C}(G, E, \Sigma_\mathrm{u})$ and in $G \parallel E$. This shows $G \parallel S \subseteq_\mathrm{v} G \parallel E$ according to Definition 5.

(iii) Let $\hat{v}, \hat{w} \in \mathrm{dom}(\mathrm{vars}(G \parallel S))$, let $(x_G, x_S, \hat{v}) \in Q^\mathrm{acc}(G \parallel S)$, let $\mu \in \Sigma_S \cap \Sigma_\mathrm{u}$, and let $(x_G, \hat{v}) \xrightarrow{\mu} (y_G, \hat{w})$ in G. Following Definition 7, it is to be shown that there exists a location y_S of S such that $(x_G, x_S, \hat{v}) \xrightarrow{\mu} (y_G, y_S, \hat{w})$ in $G \parallel S$.

Write $x_S = (x'_G, x_E)$, so that $(x_G, (x'_G, x_E), \hat{v}) = (x_G, x_S, \hat{v}) \in Q^\mathrm{acc}(G \parallel S) = Q^\mathrm{acc}(G \parallel \sup\mathcal{C}(G, E, \Sigma_\mathrm{u})) \subseteq Q^\mathrm{acc}(G \parallel G \parallel E)$, which implies $x_G = x'_G$, as G is state-deterministic. As $\mathrm{vars}'(S) = \mathrm{vars}'(G)$, it follows that $(x_G, x_E, \hat{v}) = (x'_G, x_E, \hat{v}) = (x_S, \hat{v}) \in Q^\mathrm{acc}(S) = Q^\mathrm{acc}(\sup\mathcal{C}(G, E, \Sigma_\mathrm{u})) \subseteq \hat{\Theta}_{G,E,\Sigma_\mathrm{u}} = \Theta_{G,E,\Sigma_\mathrm{u}}(\hat{\Theta}_{G,E,\Sigma_\mathrm{u}})$ by (14) and as $\hat{\Theta}_{G,E,\Sigma_\mathrm{u}}$ is a fixpoint of $\Theta_{G,E,\Sigma_\mathrm{u}}$. Then by Definition 9 there exists a location y_E of E such that $(x_G, x_E, \hat{v}) \xrightarrow{\mu} (y_G, y_E, \hat{w}) \in \hat{\Theta}_{G,E,\Sigma_\mathrm{u}}$. Then $(x_G, x_E, \hat{v}) \xrightarrow{\mu} (y_G, y_E, \hat{w})$ in $\sup\mathcal{C}(G, E, \Sigma_\mathrm{u}) = S$, which implies the claim $(x_G, x_S, \hat{v}) \xrightarrow{\mu} (y_G, y_S, \hat{w})$ in $G \parallel S$ with $y_S = (y_G, y_E)$.

(iv) Let S' be an EFSM that satisfies (ii) and (iii). Then in particular $G \parallel S' \subseteq_{\mathrm{v}} G \parallel E$ (ii), so by Definition 5 for every path

$$(x_G^0, x_S^0, \hat{w}^0) \xrightarrow{\sigma_1} (x_G^1, x_S^1, \hat{w}^1) \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_n} (x_G^n, x_S^n, \hat{w}^n) \tag{17}$$

in $G \parallel S'$, with $\hat{w}^k \in \mathrm{dom}(W)$ and $W = \mathrm{vars}(G) \cup \mathrm{vars}(S')$, there exists a path

$$(x_G^0, x_E^0, \hat{v}^0) \xrightarrow{\sigma_1} (x_G^1, x_E^1, \hat{v}^1) \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_n} (x_G^n, x_E^n, \hat{v}^n) \tag{18}$$

in $G \parallel E$, with $\hat{v}^k \in \mathrm{dom}(V)$ and $V = \mathrm{vars}(G) \cup \mathrm{vars}(E)$, such that $\hat{w}^k_{V \cap W} = \hat{v}^k_{|V \cap W}$ for $0 \leq k \leq n$. Note that the locations x_G^k are the same in both paths due to the state-determinism of G. Following Remark 1, assume without loss of generality that $\mathrm{vars}(E) \subseteq \mathrm{vars}(G)$, so that $V = \mathrm{vars}(G) \cup \mathrm{vars}(E) = \mathrm{vars}(G) \subseteq \mathrm{vars}(G) \cup \mathrm{vars}(S') = W$ and thus $\hat{v}^k = \hat{w}^k_{|V}$ for $0 \leq k \leq n$.

Let $X \subseteq Q_G \times Q_E \times \mathrm{dom}(V)$ be the set of all end states $(x_G^n, x_E^n, \hat{v}^n)$ of paths (18) in $G \parallel E$ obtained from a corresponding path (17) in $G \parallel S'$. This means that $G \parallel S' \subseteq_{\mathrm{v}} \mathrm{Acc}((G \parallel E) \upharpoonright X)$.

It is next shown that $X \subseteq \Theta_{G,E,\Sigma_{\mathrm{u}}}(X)$, i.e., X is a *post-fixpoint* of $\Theta_{G,E,\Sigma_{\mathrm{u}}}$. Let $(x_G^n, x_E^n, \hat{v}^n) \in X$ and $(x_G^n, \hat{v}^n) \xrightarrow{\mu} (x_G^{n+1}, \hat{v}^{n+1})$ in G for some $\mu \in \Sigma_{\mathrm{u}}$, $x_G^{n+1} \in Q_G$, and $\hat{v}^{n+1} \in \mathrm{dom}(V)$. As $(x_G^n, x_E^n, \hat{v}^n) \in X$, there exists a corresponding path (17) in $G \parallel S'$ with end state $(x_G^n, x_S^n, \hat{w}^n)$ such that $\hat{v}^n = \hat{w}^n_{|V}$. It follows that $(x_G^n, x_S^n, \hat{w}^n) \in Q^{\mathrm{acc}}(G \parallel S')$. Let $\hat{w}^{n+1} = \hat{v}^{n+1} \oplus \hat{w}^n$. Then it follows from $(x_G^n, \hat{v}^n) \xrightarrow{\mu} (x_G^{n+1}, \hat{v}^{n+1})$ that $(x_G^n, \hat{w}^n) \xrightarrow{\mu} (x_G^{n+1}, \hat{w}^{n+1})$, and this implies by the Σ_{u}-controllability of S' with respect to G (iii) and the state-determinism of G that $(x_G^n, x_S^n, \hat{w}^n) \xrightarrow{\mu} (x_G^{n+1}, x_S^{n+1}, \hat{w}^{n+1})$ in $G \parallel S'$ for some location x_S^{n+1} of S'. This transition extends the path (17), and by the state-determinism of G and E, it follows that the corresponding path (18) extends by $(x_G^n, x_E^n, \hat{v}^n) \xrightarrow{\mu} (x_G^{n+1}, x_E^{n+1}, \hat{v}^{n+1})$ for some location x_E^{n+1} of E. Therefore $(x_G^{n+1}, x_E^{n+1}, \hat{v}^{n+1}) \in X$ and thus $(x_G^n, x_E^n, \hat{v}^n) \in \Theta_{G,E,\Sigma_{\mathrm{u}}}(X)$ by Definition 9. As $(x_G^n, x_E^n, \hat{v}^n) \in X$ was chosen arbitrarily, it follows that $X \subseteq \Theta_{G,E,\Sigma_{\mathrm{u}}}(X)$.

It follows from the Knaster-Tarski theorem [12] that $X \subseteq \hat{\Theta}_{G,E,\Sigma_{\mathrm{u}}}$. This implies $G \parallel S' \subseteq_{\mathrm{v}} \mathrm{Acc}((G \parallel E) \upharpoonright X) \subseteq_{\mathrm{v}} \mathrm{Acc}((G \parallel E) \upharpoonright \hat{\Theta}_{G,E,\Sigma_{\mathrm{u}}}) = G \parallel \mathrm{Acc}((G \parallel E) \upharpoonright \hat{\Theta}_{G,E,\Sigma_{\mathrm{u}}}) = G \parallel \sup\mathcal{C}(G, E, \Sigma_{\mathrm{u}})$. □

3.4 Modular Synthesis Algorithm

The idea of modular synthesis (Algorithm 1) is to identify a suitable subsystem to perform synthesis and use the result to control the entire system. This is based on modularity properties, according to which synchronous composition of a state machine with another only ever

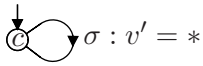

Figure 1: The EFSM chaos(σ, v).

restricts the behaviour [2]. Then controllability with respect to a part of the plant implies controllability with respect to the entire plant, as shown in Lemma 1.

EFSMs do not have this property. When EFSMs are combined in synchronous composition, new next-state variables can be added to transitions, possibly changing variables that were implicitly unchanged. To enable modular synthesis with EFSMs, one solution is to replace the parts of the system not considered in a synthesis attempt by an *abstraction* that includes all possible variable changes. This abstraction is called *chaos EFSM*.

Definition 11 Given an event σ and a variable v, the *chaos EFSM* for σ and v is defined as

$$\text{chaos}(\sigma, v) = \langle \{\sigma\}, \{c\}, \{c\}, \{(c, \sigma, v' = *, c)\} \rangle \ . \tag{19}$$

The EFSM chaos(σ, v) is shown in Figure 1. The update $v' = *$ means that v can assume any value from its domain in the next state. Formally, this update is true for all valuations, but it includes the next-state variable v' so that v is no longer implicitly unchanged.

In the synchronous composition $F_1 \parallel F_2$ of two EFSMs, some variables in F_1 may be changed by transitions in F_2. A variable v can be changed after composition of a transition in F_1 that does not mention v' with a transition in F_2 that mentions v'; or by a transition with an event that only appears in F_2. By inspection of the next-state variables on the transitions of F_2, it can be determined that certain variables are not changed in F_2, or are only changed on the occurrence of certain events. The following Lemma 5 shows how to identify the specific chaos EFSMs to capture possible variable changes in another EFSM and obtain a modularity result for EFSMs. It is the crucial argument needed to generalise Lemma 1, which is done in Lemma 6 below.

Lemma 5 Let F_1 and F_2 be two EFSMs, and let

$$C = \Big\| (\{\, \text{chaos}(\sigma, v) \mid v \in \text{vars}(F_1) \cap \text{vars}'(F_2, \sigma) \,\}) \ . \tag{20}$$

If $(x_1, x_2, \hat{v}) \xrightarrow{\sigma} (y_1, y_2, \hat{w})$ in $F_1 \parallel F_2$ then $(x_1, c, \hat{v}_{|\text{vars}(F_1)}) \xrightarrow{\sigma} (y_1, c, \hat{w}_{|\text{vars}(F_1)})$ in $F_1 \parallel C$, where c is the single location of C.

Proof. Let Σ_1, Σ_2, and Σ_C denote the event sets of F_1, F_2, and C, respectively, and write $V_1 = \text{vars}(F_1)$. Note that $\text{vars}(C) \subseteq V_1$ and thus also $V_1 = \text{vars}(F_1 \parallel C)$. Assume

$$(x_1, x_2, \hat{v}) \xrightarrow{\sigma} (y_1, y_2, \hat{w}) \tag{21}$$

in $F_1 \parallel F_2$.

If $\sigma \notin \Sigma_1 \cup \Sigma_2$ then $x_1 = y_1$ and $x_2 = y_2$, and from $\Sigma_C \subseteq \Sigma_2$ it follows that $\sigma \notin \Sigma_1 \cup \Sigma_C$. Then it is clear that $(x_1, c, \hat{v}_{|\mathrm{vars}(F_1)}) \xrightarrow{\sigma} (x_1, c, \hat{w}_{|\mathrm{vars}(F_1)}) = (y_1, c, \hat{w}_{|\mathrm{vars}(F_1)})$ in $F_1 \parallel C$.

Otherwise $F_1 \parallel F_2$ contains a transition

$$(x_1, x_2) \xrightarrow{\sigma:p} (y_1, y_2) \tag{22}$$

such that $(\hat{v} \oplus \hat{w}')(p)$ is true and $\hat{v}(v) = \hat{w}(v)$ for all $v \notin \mathrm{vars}'(p)$. C contains a single σ-transition $c \xrightarrow{\sigma:p_C} c$, where p_C is the conjunction of $v' = *$ statements over its variables $\mathrm{vars}'(p_C) = V_1 \cap \mathrm{vars}'(F_2, \sigma)$. The update p_C is true for all valuations, only its next-state variables are important. From (22) it follows that $x_1 \xrightarrow{\sigma:p_1} y_1$ in F_1 and $x_2 \xrightarrow{\sigma:p_2} y_2$ in F_2 such that $p = p_1 \wedge p_2$. (By the extended definition of the transition relation, if $\sigma \notin \Sigma_k$ for $k \in \{1, 2\}$, then $x_k = y_k$ and $p_k = \mathrm{true}$.) Then $F_1 \parallel C$ has a transition

$$(x_1, c) \xrightarrow{\sigma:p_1 \wedge p_C} (y_1, c) \ . \tag{23}$$

As $(\hat{v} \oplus \hat{w}')(p) = (\hat{v} \oplus \hat{w}')(p_1 \wedge p_2)$ is true, it follows that $(\hat{v} \oplus \hat{w}')(p_1)$ is true, and thus $(\hat{v} \oplus \hat{w}')(p_1 \wedge p_C)$ is also true.

Now consider $v \in V_1 \setminus \mathrm{vars}'(p_1 \wedge p_C)$. Then $v \in V_1$ and $v \notin \mathrm{vars}'(p_1)$ and $v \notin \mathrm{vars}'(p_C) = V_1 \cap \mathrm{vars}'(F_2, \sigma)$, and given $v \in V_1$ also $v \notin \mathrm{vars}'(F_2, \sigma) \supseteq \mathrm{vars}'(p_2)$. This means $v \notin \mathrm{vars}'(p_1 \wedge p_2) = \mathrm{vars}'(p)$ and therefore $\hat{v}(v) = \hat{w}(v)$ from above. This shows the claim $(x_1, c, \hat{v}_{|V_1}) \xrightarrow{\sigma} (y_1, c, \hat{w}_{|V_1})$. □

In a synchronous composition $F_1 \| F_2$, Lemma 5 allows F_2 to be replaced by chaos EFSMs C for variables in F_1 and events with transitions assigning to these variables in F_2. Then all transitions in $F_1 \parallel F_2$ are also possible in $F_1 \parallel C$.

Algorithm 2 uses this result to extend Algorithm 1 for EFSM plants $\mathcal{G}$ and specification E. As before, the algorithm seeks to identify suitable subsets $\mathcal{G}^i \subseteq \mathcal{G}$ of the plants and $\Sigma_{\mathrm{u}}^i \subseteq \Sigma_{\mathrm{u}}$ of the uncontrollable events, starting with no plants and trying to use the specification E as supervisor. Differently from Algorithm 1, the plants $\bar{\mathcal{G}}^i = \mathcal{G} \setminus \mathcal{G}^i$ not included at each step are replaced by appropriate chaos EFSMs $\mathcal{C}^i$. If the supervisor S^i found in the i-th iteration is controllable with respect to the abstraction $\big\|(\mathcal{G}^i) \parallel \big\|(\mathcal{C}^i)$ of the plant and all uncontrollable events, then it is returned as the result.

Otherwise line 9 extends the set Σ_{u}^i by including uncontrollable events that cause S^i to violate controllability as it is done in Algorithm 1. Then line 10 extends the plant $\mathcal{G}^i$, which like in the FSM case must include all components that may disable some uncontrollable event, however here variables must be taken into account.

Definition 12 Let $F = \langle \Sigma, Q, Q^\circ, \rightarrow \rangle$ be an EFSM. An event $\sigma \in \Sigma$ is *always enabled* at location $x \in Q$, if the disjunction $\bigvee_{i=1}^n p_i$ is valid, where $x \xrightarrow{\sigma:p_i} y_i$ for $1 \leq i \leq n$ are all the σ-transitions originating from x. Event σ is always enabled in F, if it is always enabled at every location $x \in Q$.

Algorithm 2: Modular EFSM synthesis with a single specification.

Input: normalised state-deterministic plants $\mathcal{G} = \{G_1, \ldots, G_m\}$; pure state-deterministic specification E; uncontrollable events Σ_{u}.

$\Sigma_{\mathrm{u}}^0 \leftarrow \emptyset$;
$\mathcal{G}^0 \leftarrow \emptyset$;
$\bar{\mathcal{G}}^0 \leftarrow \mathcal{G}$;
$V^0 \leftarrow \mathrm{vars}(E)$;
$\mathcal{C}^0 \leftarrow \{\, \mathrm{chaos}(\sigma, v) \mid v \in \mathrm{vars}(E) \cap \mathrm{vars}'(\mathcal{G}, \sigma) \,\}$;
$S^0 \leftarrow E$;
$i \leftarrow 0$;
while S^i *is not* Σ_{u}*-controllable with respect to* $\big\|(\mathcal{G}^i) \parallel \big\|(\mathcal{C}^i)$ **do**
  $\Sigma_{\mathrm{u}}^{i+1} \leftarrow \Sigma_{\mathrm{u}}^i \cup \{\, \mu \in \Sigma_{\mathrm{u}} \mid$ there exist $\hat{v}, \hat{w} \in \mathrm{dom}(V^i)$,
    $(x_G, c, x_S, \hat{v}) \in Q^{\mathrm{acc}}(\big\|(\mathcal{G}^i) \parallel \big\|(\mathcal{C}^i) \parallel S^i)$, and $(x_G, \hat{v}) \xrightarrow{\mu} (y_G, \hat{w})$
    in $\mathcal{G}^i$, and there is no location y_S of S^i such that
    $(x_G, c, x_S, \hat{v}) \xrightarrow{\mu} (y_G, c, y_S, \hat{w})$ in $\big\|(\mathcal{G}^i) \parallel \big\|(\mathcal{C}^i) \parallel S^i \,\}$;
  $\mathcal{G}^{i+1} \leftarrow \{\, G' \in \mathcal{G} \mid$ there exists $\mu \in \Sigma_{\mathrm{u}}^{i+1}$ such that μ is not always enabled in $G' \,\}$;
  $\bar{\mathcal{G}}^{i+1} \leftarrow \mathcal{G} \setminus \mathcal{G}^{i+1}$;
  $V^{i+1} \leftarrow \mathrm{vars}(\mathcal{G}^{i+1}) \cup \mathrm{vars}(E)$;
  $\mathcal{C}^{i+1} \leftarrow \{\, \mathrm{chaos}(\sigma, v) \mid v \in V^{i+1} \cap \mathrm{vars}'(\bar{\mathcal{G}}^{i+1}, \sigma) \,\}$;
  $S^{i+1} \leftarrow \mathrm{sup}\mathcal{C}\big(\big\|(\mathcal{G}^{i+1}) \parallel \big\|(\mathcal{C}^{i+1}), E, \Sigma_{\mathrm{u}}^{i+1}\big)$;
  $i \leftarrow i + 1$;
end
return S^i

An event is always enabled in an EFSM if it can be taken at any location, independently of variable updates. Plant EFSMs not satisfying this condition are included in the next iteration following line 10 of Algorithm 2.

It is next shown that Algorithm 2 indeed returns the least restrictive controllable supervisor. First, to confirm that the result is controllable, the following Lemma 6 extends the FSM result of Lemma 1 to EFSMs. The second plant, G_2, cannot be dropped completely and instead is replaced by appropriate chaos EFSMs C.

Lemma 6 Let G_1, G_2, and E be EFSMs, let Σ_u be a set of events, and let

$$C = \Big\|(\{\,\mathrm{chaos}(\sigma, v) \mid v \in (\mathrm{vars}(G_1) \cup \mathrm{vars}(E)) \cap \mathrm{vars}'(G_2, \sigma)\,\}) \;. \tag{24}$$

If E is Σ_u-controllable with respect to $G_1 \| C$, then E is Σ_u-controllable with respect to $G_1 \| G_2$.

Proof. Let Σ_1, Σ_2, and Σ_E denote the event sets of G_1, G_2, and E, respectively, let $\Sigma_G = \Sigma_1 \cup \Sigma_2$, and let $\Sigma_C \subseteq \Sigma_2$ denote the event set of C. Further write $V_1 = \mathrm{vars}(G_1)$, $V_2 = \mathrm{vars}(G_2)$, and $V_E = \mathrm{vars}(E)$. Assume that E is Σ_u-controllable with respect to $G_1 \parallel C$, let $\hat{v}, \hat{w} \in \mathrm{dom}(V_1 \cup V_2 \cup V_E)$, let $(x_1, x_2, x_E, \hat{v}) \in Q^\mathrm{acc}(G_1 \parallel G_2 \parallel E)$, and let $\mu \in \Sigma_E \cap \Sigma_\mathrm{u}$ such that

$$(x_1, x_2, \hat{v}) \xrightarrow{\mu} (y_1, y_2, \hat{w}) \tag{25}$$

in $G_1 \parallel G_2$. According to Definition 7, it is to be shown that

$$(x_1, x_2, x_E, \hat{v}) \xrightarrow{\mu} (y_1, y_2, y_E, \hat{w}) \tag{26}$$

in $G_1 \parallel G_2 \parallel E$ for some location y_E of E.

From $(x_1, x_2, x_E, \hat{v}) \in Q^\mathrm{acc}(G_1 \parallel G_2 \parallel E)$ it follows by Lemma 5 (with $F_1 = G_1 \parallel E$ and $F_2 = G_2$) that $(x_1, c, x_E, \hat{v}_{|V_1 \cup V_E}) \in Q^\mathrm{acc}(G_1 \| C \| E)$, where c is the single location of C. It also follows from (25) by Lemma 5 (with $F_1 = G_1$ and $F_2 = G_2$) that $(x_1, c', \hat{v}_{|V_1}) \xrightarrow{\mu} (y_1, c', \hat{w}_{|V_1})$ in $G_1 \parallel C'$, where c' is the single location of

$$C' = \Big\|(\{\,\mathrm{chaos}(\sigma, v) \mid v \in \mathrm{vars}(G_1) \cap \mathrm{vars}'(G_2, \sigma)\,\}) \;. \tag{27}$$

C differs from C' by the addition of variables that appear in E but not in G_1, however since C only contains updates of the form $v' = *$ that are always true, it also holds that $(x_1, c, \hat{v}_{|V_1 \cup V_E}) \xrightarrow{\mu} (y_1, c, \hat{w}_{|V_1 \cup V_E})$ in $G_1 \parallel C$. Since $(x_1, c, x_E, \hat{v}_{|V_1 \cup V_E}) \in Q^\mathrm{acc}(G_1 \parallel C \parallel E)$, it follows from the Σ_u-controllability of E with respect to $G_1 \| C$ that there exists a location y_E of E such that

$$(x_1, c, x_E, \hat{v}_{|V_1 \cup V_E}) \xrightarrow{\mu} (y_1, c, y_E, \hat{w}_{|V_1 \cup V_E}) \tag{28}$$

in $G_1 \| C \| E$. Also, by (25) it holds that $(x_1, x_2) \xrightarrow{\mu : p_G} (y_1, y_2)$ in $G_1 \| G_2$ such that $(\hat{v} \oplus \hat{w}')(p_G)$ is true and $\hat{v}(v) = \hat{w}(v)$ for all $v \notin \mathrm{vars}'(p_G)$. (If $\mu \notin \Sigma_G$, then $x_1 = y_1$, $x_2 = y_2$, and $p_G = \mathrm{true}$.)

From (28) and $\mu \in \Sigma_E$ it follows by Definition 6 that E has a transition $x_E \xrightarrow{\mu:p_E} y_E$ such that $(\hat{v}_{|V_1 \cup V_E} \oplus \hat{w}'_{|V_1 \cup V_E})(p_E)$ is true. Then $(\hat{v} \oplus \hat{w}')(p_E)$ is also true. Again by Definition 6, it follows that $G_1 \parallel G_2 \parallel E$ has a transition $(x_1, x_2, x_E) \xrightarrow{\mu:p_G \wedge p_E} (y_1, y_2, y_E)$. Here $(\hat{v} \oplus \hat{w}')(p_G \wedge p_E)$ is true, and for $v \notin \mathrm{vars}'(p_G \wedge p_E) \supseteq \mathrm{vars}'(p_G)$ it holds that $\hat{v}(v) = \hat{w}(v)$ as stated above. This is enough to show the claim (26). □

The following lemma is needed to lift controllability with respect to a subsystem to a larger system. It shows that any part of the plant can be added to the specification without affecting controllability. This known result for FSMs [2] extends directly to EFSMs.

Lemma 7 Let G_1, G_2, and E be EFSMs, where G_2 is state-deterministic, and let Σ_u be a set of events. If E is Σ_u-controllable with respect to $G_1 \parallel G_2$, then $E \parallel G_2$ is Σ_u-controllable with respect to $G_1 \parallel G_2$.

Proof. Let Σ_1, Σ_2, and Σ_E denote the event sets of G_1, G_2, and E, respectively. Assume that E is Σ_u-controllable with respect to $G_1 \parallel G_2$, let $V = \mathrm{vars}(G_1) \cup \mathrm{vars}(G_2) \cup \mathrm{vars}(E)$ and $\hat{v}, \hat{w} \in \mathrm{dom}(V)$, let $(x_1, x_2, x_E, \tilde{x}_2, \hat{v}) \in Q^\mathrm{acc}(G_1 \parallel G_2 \parallel E \parallel G_2)$, and let $\mu \in (\Sigma_E \cup \Sigma_2) \cap \Sigma_\mathrm{u}$ such that $(x_1, x_2, \hat{v}) \xrightarrow{\mu} (y_1, y_2, \hat{v})$ in $G_1 \parallel G_2$. According to Definition 7, it is to be shown that $(x_1, x_2, x_E, \tilde{x}_2, \hat{v}) \xrightarrow{\mu} (y_1, y_2, y_E, \tilde{y}_2, \hat{w})$ in $G_1 \parallel G_2 \parallel E \parallel G_2$ for some locations y_E of E and $\tilde{y}_2$ of G_2.

First note that, as G_2 is state-deterministic, it holds that $x_2 = \tilde{x}_2$ and $(x_1, x_2, x_E, \hat{v}) \in Q^\mathrm{acc}(G_1 \parallel G_2 \parallel E)$. Then it follows from the Σ_u-controllability of E with respect to $G_1 \parallel G_2$ that $(x_1, x_2, x_E, \hat{v}) \xrightarrow{\mu} (y_1, y_2, y_E, \hat{w})$ in $G_1 \parallel G_2 \parallel E$ for some location y_E of E. This implies $(x_1, x_2, x_E, x_2, \hat{v}) \xrightarrow{\mu} (y_1, y_2, y_E, y_2, \hat{w})$ in $G_1 \parallel G_2 \parallel E \parallel G_2$, and the claim follows with $\tilde{x}_2 = x_2$ and $\tilde{y}_2 = y_2$. □

The following lemma is the key argument to prove that the result of Algorithm 2 is least restrictive. It shows that the supervisor S^i after each iteration is an over-approximation of the least restrictive supervisor for the full plant, after composition with omitted the plants $\bar{\mathcal{G}}^i$.

Although similar to Lemma 8, the proof needs to take variables into account. The variable sets may be different between the partial plant $\mathcal{G}^i$ and the complete plant $\mathcal{G}$, and this can result in the supervisor for $\mathcal{G}$ not being controllable with respect to $\mathcal{G}^i$. Therefore, the following proof does not use behavioural inclusion and Definition 8 directly, and instead works through the synthesis iteration.

Lemma 8 At the end of every iteration of Algorithm 2 (before line 15), it holds that

$$\sup\mathcal{C}(\Big\|(\mathcal{G}), E, \Sigma_\mathrm{u}) \subseteq_\mathrm{v} \Big\|(\bar{\mathcal{G}}^{i+1}) \parallel S^{i+1} \ . \tag{29}$$

Proof. The unfolded state set of $\sup\mathcal{C}(\Big\|(\mathcal{G}), E, \Sigma_\mathrm{u})$ has state tuples $(x_G, \bar{x}_G, x_E, \hat{v})$ where $(x_G, \bar{x}_G)$ is a location of $\mathcal{G}$, separated into the components x_G of $\mathcal{G}^{i+1}$ and $\bar{x}_G$ of $\bar{\mathcal{G}}^{i+1}$, and x_E

is a location of E, and $\hat{v} \in \mathrm{dom}(V)$ with $V = \mathrm{vars}(\mathcal{G} \cup \{E\})$. The unfolded state set of $\|(\bar{\mathcal{G}}^{i+1})\|$ S^{i+1}, on the other hand, has state tuples $(\bar{x}_G, (x_G, c, x_E), \hat{v})$, where (x_G, c, x_E) is a location of $S^{i+1} = \mathrm{sup}\mathcal{C}(\|(\mathcal{G}^{i+1}) \parallel \|(\mathcal{C}^{i+1}), E, \Sigma_{\mathrm{u}}^{i+1})$ (line 14) and c is the single location of $\|(\mathcal{C}^{i+1})$. The unfolded state set of S^{i+1} contains tuples $(x_G, c, x_E, \hat{v}_{i+1})$ with $\hat{v}_{i+1} \in \mathrm{dom}(V^{i+1})$.

It is first shown that every accessible state of the global synthesis result $\mathrm{sup}\mathcal{C}(\|(\mathcal{G}), E, \Sigma_{\mathrm{u}})$ corresponds to a state of the local synthesis result, or more precisely

$$\pi(Q^{\mathrm{acc}}(\mathrm{sup}\mathcal{C}(\|(\mathcal{G}), E, \Sigma_{\mathrm{u}}))) \subseteq \hat{\Theta}_{\|(\mathcal{G}^{i+1})\|\|(\mathcal{C}^{i+1}), E, \Sigma_{\mathrm{u}}^{i+1}} \tag{30}$$

where π is an operation to erase the $\bar{\mathcal{G}}^{i+1}$ part from state tuples, defined by

$$\pi(x_G, \bar{x}_G, x_E, \hat{v}) = (x_G, c, x_E, \hat{v}_{|V^{i+1}}) \ . \tag{31}$$

Let X^j be the state sets encountered during synthesis of $S^{i+1} = \mathrm{sup}\mathcal{C}(\|(\mathcal{G}^{i+1}) \parallel \|(\mathcal{C}^{i+1}),$ $E, \Sigma_{\mathrm{u}}^{i+1})$, i.e.,

$$X^0 = Q_{\|(\mathcal{G}^{i+1})} \times \{c\} \times Q_E \times \mathrm{dom}(V^{i+1}) \ , \tag{32}$$

$$X^{j+1} = \Theta_{\|(\mathcal{G}^{i+1})\|\|(\mathcal{C}^{i+1}), E, \Sigma_{\mathrm{u}}^{i+1}}(X^j) \ . \tag{33}$$

It is shown by induction on j that

$$\pi(Q^{\mathrm{acc}}(\mathrm{sup}\mathcal{C}(\|(\mathcal{G}), E, \Sigma_{\mathrm{u}}))) \subseteq X^j \ . \tag{34}$$

For the inductive base, $j = 0$, let $(x_G, \bar{x}_G, x_E, \hat{v}) \in Q^{\mathrm{acc}}(\mathrm{sup}\mathcal{C}(\|(\mathcal{G}), E, \Sigma_{\mathrm{u}})) \subseteq Q_{\|(\mathcal{G})} \times Q_E \times$ $\mathrm{dom}(V)$. Then $\pi(x_G, \bar{x}_G, x_E, \hat{v}) = (x_G, c, x_E, \hat{v}_{|V^{i+1}}) \in Q_{\|(\mathcal{G}^{i+1})} \times \{c\} \times Q_E \times \mathrm{dom}(V^{i+1}) = X^0$.

Now assume (34) has been shown for some $j \geq 0$ and consider $j + 1$. Let

$$(x_G, \bar{x}_G, x_E, \hat{v}) \in Q^{\mathrm{acc}}(\mathrm{sup}\mathcal{C}(\|(\mathcal{G}), E, \Sigma_{\mathrm{u}})) \ . \tag{35}$$

It is to be shown that

$$\pi(x_G, \bar{x}_G, x_E, \hat{v}) = (x_G, c, x_E, \hat{v}_{|V^{i+1}}) \in X^{j+1} = \Theta_{\|(\mathcal{G}^{i+1})\|\|(\mathcal{C}^{i+1}), E, \Sigma_{\mathrm{u}}^{i+1}}(X^j) \ . \tag{36}$$

Considering Definition 9, let $\mu \in \Sigma_{\mathrm{u}}^{i+1}$ and $\hat{w}_{i+1} \in \mathrm{dom}(V^{i+1})$ such that $(x_G, c, \hat{v}_{|V^{i+1}}) \xrightarrow{\mu} (y_G, c, \hat{w}_{i+1})$ in $\|(\mathcal{G}^{i+1}) \parallel \|(\mathcal{C}^{i+1})$. It follows from Definition 6 that $x_G \xrightarrow{\mu:p} y_G$ in $\|(\mathcal{G}^{i+1})$ such that $(\hat{v}_{|V^{i+1}} \oplus \hat{w}'_{i+1})(p)$ is true, and then $(\hat{v} \oplus \hat{w}')(p)$ with $\hat{w} = \hat{w}_{i+1} \oplus \hat{v}$ is also true. Further it follows from line 11 for all EFSMs $G' \in \bar{\mathcal{G}}^{i+1}$ that $G' \notin \mathcal{G}^{i+1}$, so $\mu \in \Sigma_{\mathrm{u}}^{i+1}$ is always enabled in G' by line 10. It follows that there exists a location $\bar{y}_G$ of $\bar{\mathcal{G}}^{i+1}$ such that

$$(x_G, \bar{x}_G, \hat{v}) \xrightarrow{\mu} (y_G, \bar{y}_G, \hat{w}) \tag{37}$$

in $\|(\mathcal{G}^{i+1}) \parallel \|(\bar{\mathcal{G}}^{i+1}) = \|(\mathcal{G})$. Further it follows from (35) that $(x_G, \bar{x}_G, (x_G, \bar{x}_G, x_E), \hat{v}) \in$ $Q^{\mathrm{acc}}(\|(\mathcal{G}) \parallel \mathrm{sup}\mathcal{C}(\|(\mathcal{G}), E, \Sigma_{\mathrm{u}}))$. Then it follows from the Σ_{u}-controllability of $\mathrm{sup}\mathcal{C}(\|(\mathcal{G}), E,$

Σ_u) with respect to $\mathcal{G}$ and the state-determinism of $\mathcal{G}$ that there exists a location y_E of E such that

$$(x_G, \bar{x}_G, (x_G, \bar{x}_G, x_E), \hat{v}) \xrightarrow{\mu} (y_G, \bar{y}_G, (y_G, \bar{y}_G, y_E), \hat{w}) \tag{38}$$

in $\big\|(\mathcal{G}) \parallel \sup\mathcal{C}(\big\|(\mathcal{G}), E, \Sigma_u)$. Then also $(y_G, \bar{y}_G, (y_G, \bar{y}_G, y_E), \hat{w}) \in Q^{\mathrm{acc}}(\big\|(\mathcal{G}) \parallel \sup\mathcal{C}(\big\|(\mathcal{G}), E, \Sigma_u))$ and $(y_G, \bar{y}_G, y_E, \hat{w}) \in Q^{\mathrm{acc}}(\sup\mathcal{C}(\big\|(\mathcal{G}), E, \Sigma_u))$. It follows by inductive assumption (34) that $(y_G, c, y_E, \hat{w}_{i+1}) = \pi(y_G, \bar{y}_G, y_E, \hat{w}) \in \pi(Q^{\mathrm{acc}}(\sup\mathcal{C}(\big\|(\mathcal{G}), E, \Sigma_u)) \subseteq X^j$. Furthermore, it follows from (38) that $(x_G, x_E, \hat{v}_{|V^{i+1}}) \xrightarrow{\mu} (y_G, y_E, \hat{w}_{i+1})$ in $\big\|(\mathcal{G}^{i+1}) \parallel E$. By construction of the chaos EFSMs, and as $\big\|(\mathcal{G}^{i+1}) \parallel \big\|(\mathcal{C}^{i+1}) \parallel E$ only has variables from V^{i+1}, also

$$(x_G, c, x_E, \hat{v}_{|V^{i+1}}) \xrightarrow{\mu} (y_G, c, y_E, \hat{w}_{i+1}) \in X^j \ . \tag{39}$$

It follows by Definition 9 that $(x_G, c, x_E, \hat{v}_{|V^{i+1}}) \in \Theta_{\|(\mathcal{G}^{i+1})\|\|(\mathcal{C}^{i+1}),E,\Sigma_u^{i+1}}(X^j) = X^{j+1}$.

This completes the induction, which shows that (34) holds for all $j \geq 0$. This in turn implies (30).

Finally, to show the original claim (29), let

$$(x_G^0, \bar{x}_G^0, x_E^0, \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, \bar{x}_G^n, x_E^n, \hat{v}^n) \tag{40}$$

be a path in $\sup\mathcal{C}(\big\|(\mathcal{G}), E, \Sigma_u)$. Clearly, all states on this path are accessible, i.e., $(x_G^k, \bar{x}_G^k, x_E^k, \hat{v}^k) \in Q^{\mathrm{acc}}(\sup\mathcal{C}(\big\|(\mathcal{G}), E, \Sigma_u))$ for all $0 \leq k \leq n$, and therefore by (30),

$$(x_G^k, c, x_E^k, \hat{v}^k_{|V^{i+1}}) = \pi(x_G^k, \bar{x}_G^k, x_E^k, \hat{v}^k) \in \hat{\Theta}_{\|(\mathcal{G}^{i+1})\|\|(\mathcal{C}^{i+1}),E,\Sigma_u^{i+1}} \ . \tag{41}$$

Furthermore, the path (40) also exists in $\big\|(\mathcal{G}) \parallel E$, which implies by Lemma 5 that $(x_G^0, c, x_E^0, \hat{v}^0_{|V^{i+1}}) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, c, x_E^n, \hat{v}^n_{|V^{i+1}})$ is a path in $\big\|(\mathcal{G}^{i+1}) \parallel \big\|(\mathcal{C}^{i+1}) \parallel E$, and by (41) also in $\sup\mathcal{C}(\big\|(\mathcal{G}^{i+1}) \parallel \big\|(\mathcal{C}^{i+1}), E, \Sigma_u^{i+1}) = S^{i+1}$. Combining this with (40), the path

$$(\bar{x}_G^0, (x_G^0, c, x_E^0), \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (\bar{x}_G^n, (x_G^n, c, x_E^n), \hat{v}^n) \tag{42}$$

in $\big\|(\bar{\mathcal{G}}^{i+1}) \parallel S^{i+1}$ is obtained. (Variables that do not appear as next-state variable on a transition in (42) do not appear at all in $\mathcal{G}^{i+1}$ or E as the construction of S^{i+1} by restriction (Definition 3) requires all variables to appear as next-state variables on all transitions, nor in $\bar{\mathcal{G}}^{i+1}$. These variables must remain unchanged according to (40).) The claim (29) follows by Definition 5. □

The following Proposition 9 combines the above lemmas to confirm that Algorithm 2 correctly returns a least restrictive supervisor. As the algorithm clearly terminates because the set Σ_u^i of uncontrollable events increases with each iteration, this completes the correctness proof of Algorithm 2.

Proposition 9 Upon termination of Algorithm 2, S^i is a supremal supervisor for E with respect to $\|(\mathcal{G})$.

Proof. It is to be shown that S^i and $G = \|(\mathcal{G})$ satisfy conditions (i)–(iv) in Definition 8.

(i) In the case $i = 0$, it holds that $\mathrm{vars}(S^0) = \mathrm{vars}(E) \subseteq \mathrm{vars}(G) \cup \mathrm{vars}(E)$ by line 6 of Algorithm 2. Otherwise, it follows that $\mathrm{vars}(S^{i+1}) = \mathrm{vars}(\mathrm{sup}\mathcal{C}(\|(\mathcal{G}^{i+1}) \parallel \|(\mathcal{C}^{i+1}), E, \Sigma_{\mathrm{u}}^{i+1})) \subseteq \mathrm{vars}(\mathcal{G}^{i+1}) \cup \mathrm{vars}(\mathcal{C}^{i+1}) \cup \mathrm{vars}(E) \subseteq \mathrm{vars}(\mathcal{G}^{i+1}) \cup \mathrm{vars}(\bar{\mathcal{G}}^{i+1}) \cup \mathrm{vars}(E) = \mathrm{vars}(\mathcal{G}) \cup \mathrm{vars}(E)$ from lines 11–14.

(ii) In the case $i = 0$, it holds that $\|(\mathcal{G}) \parallel S^0 = \|(\mathcal{G}) \parallel E$ by line 6 of Algorithm 2.

Otherwise consider a path in $\|(\mathcal{G}) \parallel S^{i+1}$,

$$(x_G^0, \bar{x}_G^0, x_S^0, \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, \bar{x}_G^n, x_S^n, \hat{v}^0) \tag{43}$$

where the locations of $\|(\mathcal{G})$ are decomposed into the parts x_G^k and $\bar{x}_G^k$ for $\|(\mathcal{G}^{i+1})$ and $\|(\bar{\mathcal{G}}^{i+1})$, respectively. The locations of $S^{i+1} = \mathrm{sup}\mathcal{C}(\|(\mathcal{G}^{i+1}) \parallel \|(\mathcal{C}^{i+1}), E, \Sigma_{\mathrm{u}}^{i+1})$ are further decomposed into $x_S^k = (x_G^k, c, x_E^k)$, where the x_G^k are the same as in (43) due to the state-determinism of $\mathcal{G}$. By Lemma 5,

$$(x_G^0, c, (x_G^0, c, x_E^0), \hat{v}^0_{|V^{i+1}}) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, c, (x_G^n, c, x_E^n), \hat{v}^n_{|V^{i+1}}) \tag{44}$$

is a path in $\|(\mathcal{G}^{i+1}) \parallel \|(\mathcal{C}^{i+1}) \parallel S^{i+1}$. As $S^{i+1} = \mathrm{sup}\mathcal{C}(\|(\mathcal{G}^{i+1}) \parallel \|(\mathcal{C}^{i+1}), E, \Sigma_{\mathrm{u}}^{i+1})$ is a supremal supervisor for E with respect to $\|(\mathcal{G}^{i+1}) \| \|(\mathcal{C}^{i+1})$ by Proposition 4, it holds by Definition 8 (ii) that $\|(\mathcal{G}^{i+1}) \parallel \|(\mathcal{C}^{i+1}) \parallel S^{i+1} \subseteq_{\mathrm{v}} \|(\mathcal{G}^{i+1}) \parallel \|(\mathcal{C}^{i+1}) \parallel E$. Then according to Definition 5, there exists a path

$$(x_G^0, c, x_E^0, \hat{v}^0_{|V^{i+1}}) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, c, x_E^n, \hat{v}^n_{|V^{i+1}}) \tag{45}$$

in $\|(\mathcal{G}^{i+1}) \parallel \|(\mathcal{C}^{i+1}) \parallel E$, where the locations x_G^k of $\|(\mathcal{G}^{i+1})$ and x_E^k of E are again the same due to state-determinism. Combining (43) and (45), it is possible to construct a path

$$(x_G^0, \bar{x}_G^0, x_E^0, \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, \bar{x}_G^n, x_E^n, \hat{v}^n) \tag{46}$$

in $\|(\mathcal{G}) \parallel E$. (Variables that do not appear as next-state variables on the i-th transition in (46), do not appear at all as next-state variables associated with σ_i in $\|(\mathcal{G})$ due to normalisation (Definition 2), and therefore also not in $\|(\mathcal{C}^{i+1})$ or on the corresponding transition in S^{i+1}. These variables must remain unchanged between $\hat{v}^i$ and $\hat{v}^{i+1}$ according to (43).) This shows $\|(\mathcal{G}) \parallel S^{i+1} \subseteq_{\mathrm{v}} \|(\mathcal{G}) \parallel E$.

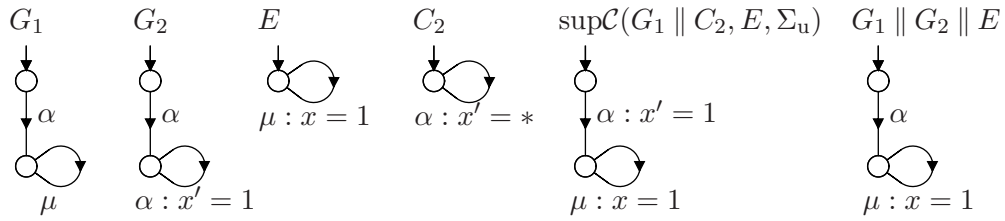

Figure 2: Issue arising from a plant that is not normalised.

(iii) By the loop-entry condition (line 8 of Algorithm 2), it is clear that S^i is Σ_u-controllable with respect to $\|(\mathcal{G}^i) \parallel \|(\mathcal{C}^i)$ upon termination of the loop. It follows by Lemma 6 (with $G_1 = \|(\mathcal{G}^i)$, $G_2 = \|(\bar{\mathcal{G}}^i)$, and $E = S^i$; note $\text{vars}(\mathcal{G}^i) \cup \text{vars}(S^i) = \text{vars}(\mathcal{G}^i) \cup \text{vars}(\mathcal{G}^i) \cup \text{vars}(E) = \text{vars}(\mathcal{G}^i) \cup \text{vars}(E) = V^i$, so C in Lemma 6 becomes the same as $\|(\mathcal{C}^i))$ that S^i is Σ_u-controllable with respect to $\|(\mathcal{G}^i) \parallel \|(\bar{\mathcal{G}}^i) = \|(\mathcal{G})$.

(iv) Let S' be an EFSM that satisfies (ii) and (iii). Then $\|(\mathcal{G}) \| S' \subseteq_v \|(\mathcal{G}) \| \sup\mathcal{C}(\|(\mathcal{G}), E, \Sigma_u)$ by Proposition 4. In the case $i = 0$, it holds that $\|(\mathcal{G}) \| S' \subseteq_v \|(\mathcal{G}) \| \sup\mathcal{C}(\|(\mathcal{G}), E, \Sigma_u) = \sup\mathcal{C}(\|(\mathcal{G}), E, \Sigma_u) \subseteq_v \|(\mathcal{G}) \parallel E = \|(\mathcal{G}) \parallel S^0$ by state-determinism of $\mathcal{G}$ and from lines 3 and 6. Otherwise it follows from Lemma 8 and again by state-determinism of $\mathcal{G}$ that $\|(\mathcal{G}) \| S' \subseteq_v \|(\mathcal{G}) \| \sup\mathcal{C}(\|(\mathcal{G}), E, \Sigma_u) = \sup\mathcal{C}(\|(\mathcal{G}), E, \Sigma_u) \subseteq_v \|(\bar{\mathcal{G}}^{i+1}) \| S^{i+1} = \|(\bar{\mathcal{G}}^{i+1}) \| \sup\mathcal{C}(\|(\mathcal{G}^{i+1}) \| \|(\mathcal{C}^{i+1}), E, \Sigma_u^{i+1}) = \|(\bar{\mathcal{G}}^{i+1}) \| \|(\mathcal{G}^{i+1}) \| \sup\mathcal{C}(\|(\mathcal{G}^{i+1}) \| \|(\mathcal{C}^{i+1}), E, \Sigma_u^{i+1}) = \|(\mathcal{G}) \parallel S^{i+1}$. □

The above proof of condition (ii) requires that the plant is normalised according to Definition 2. The following example shows that normalisation is required.

Example 1 Consider the EFSM system consisting of plants $\mathcal{G} = \{G_1, G_2\}$ and specification E as shown in Figure 2. Variable x has domain $\text{dom}(x) = \{0, 1\}$ and initial value $x^\circ = 0$, event α is controllable, while μ is uncontrollable. G_2 is not normalised because of the two updates associated with event α, namely $x' = 1$ with $\text{vars}'(x' = 1) = \{x\}$ and true with $\text{vars}'(\text{true}) = \emptyset$. If synthesis is considered for a subsystem consisting of G_1 and E, then G_2 is replaced by the chaos EFSM $C_2 = \text{chaos}(\alpha, x)$ also shown in the figure. In the composition of G_1 and C_2, the variable x can change arbitrarily on event α, so synthesis constrains this to a next-state value of 1 to satisfy the specification. This results in the supervisor $\sup\mathcal{C}(G_1 \parallel C_2, E, \Sigma_u)$. It is behaviourally included in the local system $G_1 \parallel C_2 \parallel E$, but not in the complete system $G_1 \parallel G_2 \parallel E$ where the variable x remains implicitly unchanged on the α-transition. Synthesis for the complete system has to disable α entirely, as it is not possible to change x from its initial value 0 before the uncontrollable event μ occurs.

3.5 Multiple Specifications

If an FSM system contains more than one specification, then controllability can be verified for each specification separately, and it is also known that a least restrictive supervisor can be obtained by combining the least restrictive supervisors obtained for the individual specifications [2]. Under the assumption of pure specifications, these results extend directly to EFSMs.

Lemma 10 Let G be a normalised state-deterministic EFSM, let E_1 and E_2 be pure EFSMs, and let Σ_u be a set of events. Then

$$G \parallel \sup\mathcal{C}(G, E_1 \parallel E_2, \Sigma_\mathrm{u}) \subseteq_\mathrm{v} G \parallel \sup\mathcal{C}(G, E_1, \Sigma_\mathrm{u}) \; ; \tag{47}$$

$$G \parallel \sup\mathcal{C}(G, E_1 \parallel E_2, \Sigma_\mathrm{u}) \subseteq_\mathrm{v} G \parallel \sup\mathcal{C}(G, E_2, \Sigma_\mathrm{u}) \; . \tag{48}$$

Proof. Write $E = E_1 \parallel E_2$. Note that $S = \sup\mathcal{C}(G, E, \Sigma_\mathrm{u})$ is a supremal supervisor for E with respect to G by Proposition 4. To show the claim (47), following Definition 8, it is enough to show that S satisfies conditions (ii) and (iii) in Definition 8 (with $E = E_1$).

(ii) Let

$$(x_G^0, x_S^0, \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, x_S^n, \hat{v}^n) \tag{49}$$

be a path in $G \parallel S$ where $\hat{v}^k \in \mathrm{dom}(\mathrm{vars}(G \parallel S))$. As $G \parallel S \subseteq_\mathrm{v} G \parallel E$ (ii), there exists a path

$$(x_G^0, x_E^0, \hat{w}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, x_E^n, \hat{w}^n) \tag{50}$$

in $G \parallel E$ where $\hat{w}^k \in \mathrm{dom}(\mathrm{vars}(G \parallel E))$, and $\hat{v}^k_{|\mathrm{vars}(G\parallel S)} = \hat{w}^k_{|\mathrm{vars}(G\parallel E)}$ for $1 \leq k \leq n$; note that the locations x_G^k of G are the same due to state-determinism. Decompose the locations of E into the components of E_1 and E_2 by writing $x_E^k = (x_{E1}^k, x_{E2}^k)$. As $\mathrm{vars}'(E_2) = \emptyset$, it follows that

$$(x_G^0, x_{E1}^0, \hat{w}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, x_{E1}^n, \hat{w}^n) \tag{51}$$

must be a path in $G \parallel E_1$. As the path (49) was chosen arbitrarily, it follows from Definition 5 that $G \parallel S \subseteq_\mathrm{v} G \parallel E_1$.

(iii) S is Σ_u-controllable with respect to G, because $S = \sup\mathcal{C}(G, E, \Sigma_\mathrm{u})$ is a supremal supervisor for E with respect to G (iii).

This shows (47). The proof for (48) is analogous. □

Proposition 11 Let G be a state-deterministic normalised EFSM, let E_1 and E_2 be pure EFSMs, and let Σ_u be a set of events. Let S_1 and S_2 be supremal supervisors for E_1 and E_2 with respect to G, respectively. Then $S_1 \parallel S_2$ is a supremal supervisor for $E_1 \parallel E_2$ with respect to G.

Proof. It is to be shown that $S = S_1 \parallel S_2$ satisfies conditions (i)–(iv) in Definition 8 (with $E = E_1 \parallel E_2$).

(i) Clearly, $\mathrm{vars}(S) = \mathrm{vars}(S_1) \cup \mathrm{vars}(S_2) \subseteq \mathrm{vars}(G) \cup \mathrm{vars}(E_1) \cup \mathrm{vars}(G) \cup \mathrm{vars}(E_2) = \mathrm{vars}(G) \cup \mathrm{vars}(E)$ as S_1 and S_2 are supremal supervisors for E_1 and E_2, respectively.

(ii) Write $V = \mathrm{vars}(G)$. Let

$$(x_G^0, x_{S1}^0, x_{S2}^0, \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, x_{S1}^n, x_{S2}^n, \hat{v}^n) \tag{52}$$

be a path in $G \parallel S = G \parallel S_1 \parallel S_2$ where $\hat{v}^k \in \mathrm{dom}(\mathrm{vars}(G \parallel S))$ for $0 \leq k \leq n$. Following Remark 1, assume without loss of generality that $\mathrm{vars}(E_1), \mathrm{vars}(E_2) \subseteq \mathrm{vars}(G)$, which by (i) implies $\mathrm{vars}(G \parallel S) = \mathrm{vars}(G) \cup \mathrm{vars}(S_1) \cup \mathrm{vars}(S_2) \subseteq \mathrm{vars}(G) \cup \mathrm{vars}(G) \cup \mathrm{vars}(E_1) \cup \mathrm{vars}(G) \cup \mathrm{vars}(E_2) = \mathrm{vars}(G) = V$, i.e., $\hat{v}^k \in \mathrm{dom}(V)$ for $0 \leq k \leq n$.

It is shown by induction on n that

$$(x_G^0, x_{S1}^0, \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, x_{S1}^n, \hat{v}^n) \tag{53}$$

$$(x_G^0, x_{S2}^0, \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, x_{S2}^n, \hat{v}^n) \tag{54}$$

are paths in $G \parallel S_1$ and $G \parallel S_2$. This clearly is the case for $n = 0$. Now assume paths (53) and (54) of length $n \geq 0$ have been constructed, and consider a path (52) of length $n + 1$,

$$(x_G^0, x_{S1}^0, x_{S2}^0, \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, x_{S1}^n, x_{S2}^n, \hat{v}^n) \xrightarrow{\sigma_{n+1}} (x_G^{n+1}, x_{S1}^{n+1}, x_{S2}^{n+1}, \hat{v}^{n+1}) \ . \tag{55}$$

The final transition of (55) implies by Definition 6 that $x_G^n \xrightarrow{\sigma_{n+1}:p_G} x_G^{n+1}$ in G and $x_{S1}^n \xrightarrow{\sigma_{n+1}:p_{S1}} x_{S1}^{n+1}$ in S_1 and $x_{S2}^n \xrightarrow{\sigma_{n+1}:p_{S2}} x_{S2}^{n+1}$ in S_2, where $(\hat{v}^n \oplus (\hat{v}^{n+1})')(p_G)$, $(\hat{v}^n \oplus (\hat{v}^{n+1})')(p_{S1})$, and $(\hat{v}^n \oplus (\hat{v}^{n+1})')(p_{S2})$ are all true, and

$$\hat{v}^n(v) = \hat{v}^{n+1}(v) \qquad \text{for all } v \in V \setminus \mathrm{vars}'(p_G \wedge p_{S1} \wedge p_{S2}) \ . \tag{56}$$

Again by Definition 6, there exists a transition

$$(x_G^n, x_{S1}^n) \xrightarrow{\sigma_{n+1}:p_G \wedge p_{S1}} (x_G^{n+1}, x_{S1}^{n+1}) \tag{57}$$

in $G \parallel S_1$ such that $(\hat{v}^n \oplus (\hat{v}^{n+1})')(p_G \wedge p_{S1})$ is true. It remains to be shown that variables in $V_1 = V \setminus \mathrm{vars}'(p_G \wedge p_{S1})$ are unchanged between $\hat{v}^n$ and $\hat{v}^{n+1}$. However, so far it only follows that $(x_G^n, x_{S1}^n, \hat{v}^n) \xrightarrow{\sigma_{n+1}} (x_G^{n+1}, x_{S1}^{n+1}, \hat{v}_{|V_1}^n \oplus \hat{v}^{n+1})$ in $G \parallel S_1$. This transition extends the path (53) from the inductive assumption. Given that S_1 is a supremal supervisor for E_1, i.e., $G \parallel S_1 \subseteq_{\mathrm{v}} G \parallel E_1$ by Definition 8 (ii), it follows by Definition 5 that there exists a path

$$(x_G^0, x_{E1}^0, \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, x_{E1}^n, \hat{v}^n) \xrightarrow{\sigma_{n+1}} (x_G^{n+1}, x_{E1}^{n+1}, \hat{v}_{|V_1}^n \oplus \hat{v}^{n+1}) \ . \tag{58}$$

in $G \parallel E_1$. The locations of G are unchanged from (55) due to state-determinism. Then $G \parallel E_1$ must contain a transition $(x_G^n, x_{E1}^n) \xrightarrow{\sigma_{n+1}:p_G \wedge p_{E1}} (x_G^{n+1}, x_{E1}^{n+1})$, such that variables outside of $\mathrm{vars}'(p_G \wedge p_{E1}) = \mathrm{vars}'(p_G)$ (note p_{E1} is an update from the pure EFSM E_1) are unchanged on the last transition of (58), i.e.,

$$\hat{v}^n_{|V \setminus \mathrm{vars}'(p_G)} = (\hat{v}^n_{|V_1} \oplus \hat{v}^{n+1})_{|V \setminus \mathrm{vars}'(p_G)} \ . \tag{59}$$

By a symmetric argument, it can be shown that

$$\hat{v}^n_{|V \setminus \mathrm{vars}'(p_G)} = (\hat{v}^n_{|V_2} \oplus \hat{v}^{n+1})_{|V \setminus \mathrm{vars}'(p_G)} \ , \tag{60}$$

where $V_2 = V \setminus \mathrm{vars}'(p_G \wedge p_{S2})$. Now let $v \in V_1$ and consider two cases.

- $v \notin \mathrm{vars}'(p_{S2})$. Then it follows from $v \in V_1 = V \setminus \mathrm{vars}'(p_G \wedge p_{S1})$ that $v \notin \mathrm{vars}'(p_G \wedge p_{S1} \wedge p_{S2})$ and thus $\hat{v}^n(v) = \hat{v}^{n+1}(v)$ by (56).
- $v \in \mathrm{vars}'(p_{S2})$. Note that $v \in V_1 = V \setminus \mathrm{vars}'(p_G \wedge p_{S1}) \subseteq V \setminus \mathrm{vars}'(p_G)$ and $v \notin V \setminus \mathrm{vars}'(p_{S2}) \supseteq V \setminus \mathrm{vars}'(p_{S2} \wedge p_G) = V_2$, and therefore it follows from (60) that $\hat{v}^n(v) = \hat{v}^n_{|V \setminus \mathrm{vars}'(p_G)}(v) = (\hat{v}^n_{|V_2} \oplus \hat{v}^{n+1})_{|V \setminus \mathrm{vars}'(p_G)}(v) = (\hat{v}^n_{|V_2} \oplus \hat{v}^{n+1})(v) = \hat{v}^{n+1}(v)$.

Thus, $\hat{v}^n(v) = \hat{v}^{n+1}(v)$ for all $v \in V_1 = V \setminus \mathrm{vars}'(p_G \wedge p_{S1})$, so it follows from (57) that

$$(x_G^n, x_{S1}^n, \hat{v}^n) \xrightarrow{\sigma_{n+1}} (x_G^{n+1}, x_{S1}^{n+1}, \hat{v}^{n+1}) \tag{61}$$

in $G \parallel S_1$. Thus, the path (53) also exists for $n+1$. The path (54) is extended by a symmetric argument, completing the induction.

Therefore, the paths (53) and (54) exist in $G \| S_1$ and $G \| S_2$. Given that S_1 is a supervisor for E_1, i.e., $G \parallel S_1 \subseteq_{\mathrm{v}} G \parallel E_1$ by Definition 8 (ii), and likewise $G \parallel S_2 \subseteq_{\mathrm{v}} G \parallel E_2$, by Definition 5 there also exist paths

$$(x_G^0, x_{E1}^0, \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, x_{E1}^n, \hat{v}^n) \tag{62}$$

$$(x_G^0, x_{E2}^0, \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, x_{E2}^n, \hat{v}^n) \tag{63}$$

in $G \parallel E_1$ and $G \parallel E_2$. As E_1 and E_2 are pure, it follows that

$$(x_G^0, x_{E1}^0, x_{E_2}^0, \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_n} (x_G^n, x_{E1}^n, x_{E_2}^n, \hat{v}^n) \tag{64}$$

is a path in $G \parallel E_1 \parallel E_2 = G \parallel E$. Since the path (52) was chosen arbitrarily, it follows from Definition 5 that $G \parallel S \subseteq_{\mathrm{v}} G \parallel E$.

(iii) Let $\hat{v}, \hat{w} \in \mathrm{dom}(\mathrm{vars}(G \parallel S))$, let $(x_G, x_S, \hat{v}) \in Q^{\mathrm{acc}}(G \parallel S)$, let $\mu \in \Sigma_S \cap \Sigma_{\mathrm{u}}$, and let $(x_G, \hat{v}) \xrightarrow{\mu} (y_G, \hat{w})$ in G. Following Definition 7, it is to be shown that there exists a location y_S of S such that $(x_G, x_S, \hat{v}) \xrightarrow{\mu} (y_G, y_S, \hat{w})$ in $G \parallel S$.

Write $x_S = (x_{S1}, x_{S2})$. As $(x_G, x_{S1}, x_{S2}, \hat{v}) = (x_G, x_S, \hat{v}) \in Q^{\mathrm{acc}}(G \parallel S) = Q^{\mathrm{acc}}(G \parallel S_1 \parallel S_2)$, it follows from Lemma 5 (with $F_1 = G \parallel S_1$ and $F_2 = S_2$) that $(x_G, x_{S1}, c, \hat{v}) \in Q^{\mathrm{acc}}(G \| S_1 \| C)$. Given that $\mathrm{vars}'(C) \subseteq \mathrm{vars}'(S_2) = \mathrm{vars}'(G \| E_2) = \mathrm{vars}'(G) \cup \mathrm{vars}'(E_2) = \mathrm{vars}'(G) \subseteq \mathrm{vars}'(G \| S_1)$, as $\mathrm{vars}'(E_2) = \emptyset$ by assumption, it also holds that $(x_G, x_{S1}, \hat{v}) \in Q^{\mathrm{acc}}(G \parallel S_1)$. Then by the Σ_{u}-controllability of S_1 with respect to G, there exists a location y_{S1} of S_1 such that $(x_G, x_{S1}, \hat{v}) \xrightarrow{\mu} (y_G, y_{S1}, \hat{w})$ in $G \parallel S_1$. Likewise, there exists a location y_{S2} of S_2 such that $(x_G, x_{S2}, \hat{v}) \xrightarrow{\mu} (y_G, y_{S2}, \hat{w})$ in $G \parallel S_2$. Then it also follows that $(x_G, x_S, \hat{v}) = (x_G, x_{S1}, x_{S2}, \hat{v}) \xrightarrow{\mu} (y_G, y_{S1}, y_{S2}, \hat{w})$ in $G \parallel S_1 \parallel S_2 = G \parallel S$.

(iv) Let S' be an EFSM that satisfies (ii) and (iii). Then $G \parallel S' \subseteq_{\mathrm{v}} G \parallel \sup\mathcal{C}(G, E, \Sigma_{\mathrm{u}}) \subseteq_{\mathrm{v}} G \parallel \sup\mathcal{C}(G, E_j, \Sigma_{\mathrm{u}}) \subseteq_{\mathrm{v}} G \parallel S_j$, for $j \in \{1, 2\}$, by Proposition 4 and Lemma 10. Now let

$$(x_G^0, x_S^0, \hat{v}^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_\eta} (x_G^n, x_S^n, \hat{v}^n) \tag{65}$$

be a path in $G \parallel S'$. As $G \parallel S' \subseteq_{\mathrm{v}} G \parallel S_1$ and $G \parallel S' \subseteq_{\mathrm{v}} G \parallel S_2$, there exist paths

$$(x_G^0, x_{S1}^0, \hat{v}_1^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_\eta} (x_G^n, x_{S1}^n, \hat{v}_1^n) \tag{66}$$

$$(x_G^0, x_{S2}^0, \hat{v}_2^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_\eta} (x_G^n, x_{S2}^n, \hat{v}_2^n) \tag{67}$$

in $G \| S_1$ and $G \| S_2$, with $(\hat{v}^k)_{|\mathrm{vars}(G\|S') \cap \mathrm{vars}(G\|S_j)} = (\hat{v}_j^k)_{|\mathrm{vars}(G\|S') \cap \mathrm{vars}(G\|S_j)}$ for $j \in \{1, 2\}$ and $1 \leq k \leq n$. Note that $(\hat{v}^k)_{|\mathrm{vars}(G)} = (\hat{v}_j^k)_{|\mathrm{vars}(G)}$. Then

$$(x_G^0, x_{S1}^0, x_{S2}^0, \hat{v}_1^0 \oplus \hat{v}_2^0) \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_\eta} (x_G^n, x_{S1}^n, x_{S2}^n, \hat{v}_1^n \oplus \hat{v}_2^n) \tag{68}$$

is a path in $G \| S_1 \| S_2 = G \| S$, with $(\hat{v}_k)_{|\mathrm{vars}(G\|S') \cap \mathrm{vars}(G\|S)} = (\hat{v}_1^k \oplus \hat{v}_2^k)_{|\mathrm{vars}(G\|S') \cap \mathrm{vars}(G\|S)}$ for $0 \leq k \leq n$. As the path (65) was chosen arbitrarily, it follows from Definition 5 that $G \parallel S' \subseteq_{\mathrm{v}} G \parallel S$. □

Given sets of plant EFSMs $\mathcal{G}$ and specifications EFSMs $\mathcal{E}$, by Propositions 9 and 11 synthesis can be performed separately for each specification $E_j \in \mathcal{E}$ with respect to $\mathcal{G}$ using Algorithm 2, and the synchronous composition of the resulting supervisors $S_j = \sup\mathcal{C}(\|(\mathcal{G}), E_j, \Sigma_{\mathrm{u}})$ gives the least restrictive supervisor for the combined specification $\|(\mathcal{E})$ and plant $\|(\mathcal{G})$. This approach is shown in Algorithm 3.

4 Manufacturing system example

Figure 3 shows a manufacturing system with material feedback [13]. The system consists of two machines, M_1 and M_2, linked by one-place buffers B_1 and B_2. Buffer B_1 receives external workpieces by event s. Machine M_1 removes workpieces from B_1 (event s_1), manufactures and puts them in B_2 (event f_1), where a quality test is performed. The test determines

Algorithm 3: Modular EFSM synthesis with multiple specifications.

Input: normalised state-deterministic plants $\mathcal{G} = \{G_1, \ldots, G_m\}$; pure specifications $\mathcal{E} = \{E_1, \ldots, E_l\}$; uncontrollable events Σ_u.

```
S ← ∅;
foreach E_j ∈ E do
    Calculate S_j using Algorithm 2 with E = E_j;
    S ← S ∪ {S_j};
end
return S
```

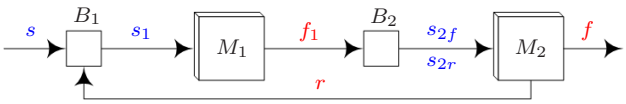

Figure 3: Manufacturing system with material feedback [13].

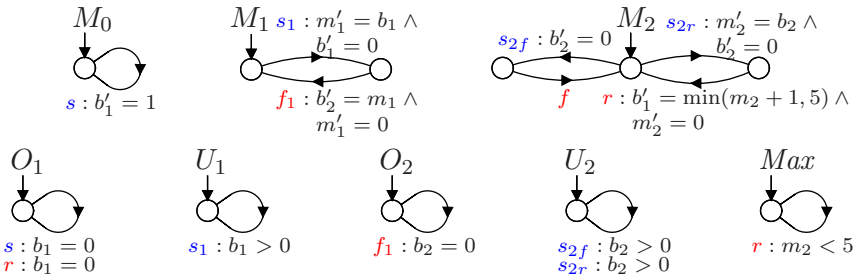

Figure 4: EFSM model of manufacturing system.

the operation to be performed by M_2 (s_{2f} or s_{2r}), which leads to a release of the workpiece (event f) or its return to B_1 for rework (event r). Events f_1, f, and r are uncontrollable, the others are controllable. The control objective is to avoid overflow and underflow of the buffers, and to ensure that workpieces pass through the system at most five times.

Figure 4 shows an EFSM model of this system. The variables b_1, m_1, b_2, and m_2 with domain $\{0, 1, 2\}$ and initial value 0 record the contents of machines and buffers. A value of 0 indicates that the machine or buffer is empty, while a value of $k > 0$ indicates the presence of a workpiece that has been placed k times into buffer B_1.

Plants M_0, M_1, and M_2 model the behaviour of the machines and buffers. A first unload to B_1 (transition s in M_0) sets the variable b_1 to 1, indicating presence of a workpiece in its first cycle. Loading a workpiece from B_1 to M_1 (transition s_1 in M_1) transfers the value from b_1 to m_1 and resets b_1 to 0, as B_1 is again empty. Likewise, a transition with f_1 assigns the

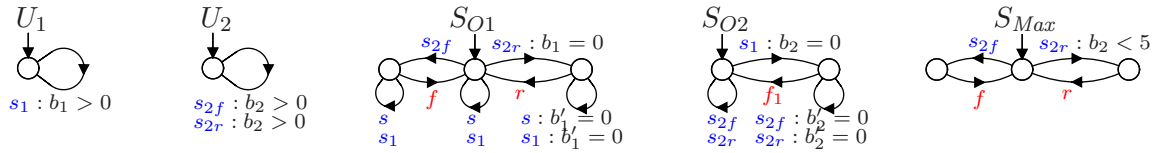

Figure 5: Synthesised supervisors for manufacturing system.

value of m_1 to b_2 and resets m_1 to 0, etc. When a workpiece is returned to B_1 on event r, the condition $b_1' = \min(m_2 + 1, 5)$ in M_2 assigns to b_1 the value that identifies the next work cycle. The transition is also defined when $m_2 = 5$, i.e., the workpiece is already in its last cycle, but this case is ruled out by specification Max.

Specifications O_1, U_1, O_2, and U_2 model the avoidance of overflow and underflow. For example, O_1 specifies that a workpiece can only be added to buffer B_1 on event s or r when it is empty, $b_1 = 0$. Specification Max restricts the number of cycles per workpiece.

To obtain a modular supervisor, Algorithm 2 is invoked for each specification separately. Specifications U_1 and U_2 contain only controllable events, so they are naturally controllable and Algorithm 2 terminates without entering the loop. These specifications can serve as supervisors directly, e.g., U_1 prevents loading of a workpiece from B_1 when B_1 is empty.

For specification O_1, the first iteration of Algorithm 2 without plants, $\mathcal{G}^0 = \emptyset$, uses chaos EFSMs for variable b_1, which appears in O_1 and is changed by the plant on events s, s_1, and r, i.e., $\mathcal{C}^0 = \{\text{chaos}(s, b_1), \text{chaos}(s_1, b_1), \text{chaos}(r, b_1)\}$. It turns out that O_1 is not Σ_u-controllable with respect to $\|(\mathcal{C}^0)$, because O_1 prevents the uncontrollable event r when $b_1 \neq 0$. Therefore, r is considered as uncontrollable in the next iteration, $\Sigma_u^1 = \{r\}$, so plant M_2 that disables r is added, $\mathcal{G}^1 = \{M_2\}$. Among the variables in M_2 and O_1, $V^1 = \{b_1, b_2, m_2\}$, only b_1 and b_2 may be changed by the remaining plants M_0 and M_1, so that $\mathcal{C}^1 = \{\text{chaos}(s, b_1), \text{chaos}(s_1, b_1), \text{chaos}(f_1, b_2)\}$. Synthesis gives the supervisor $S_{O1} = \sup\mathcal{C}(M_2 \parallel \|(\mathcal{C}^1), O_1, \{r\})$ shown in Figure 5, which is Σ_u-controllable with respect to $M_2 \parallel \|(\mathcal{C}^1)$. Algorithm 2 stops here.

The figure only shows updates added during synthesis, a computed supervisor may contain more updates. The update on s_{2r} in S_{O1} avoids overflow of B_1 by preventing M_2 from loading a workpiece to be reworked unless B_1 is empty. The update on s contradicts plant M_0 and thereby disables s, i.e., the loading of a new workpiece to B_1, while another is being returned. The update on s_1 is redundant due to plant M_1, but appears here as M_1 was not included in the partial synthesis.

Synthesis for specifications O_2 and Max also terminates after the first iteration, producing the supervisors S_{O1} and S_{Max} in Figure 5.

Table 1 shows an overview of the synthesis procedure for this example and the numbers of states encountered at the end of each iteration. Overall, modular synthesis never composes more than two of the EFSMs from Figure 4 and gives five supervisor components with 1–3

Table 1: Synthesis Statistics for Example.

Specifications E	Plants $\mathcal{G}^n$	Variables V^n	Uncont. Σ_{u}^n	Explored states $\left\|\left(\mathcal{G}^n\right) \parallel \left\|\left(\mathcal{C}^n\right) \parallel E\right.\right.$	Sup. locations S^n
Modular Synthesis					
O_1	M_2	b_1, b_2, m_2	r	288	3
U_1	—	b_1	—	3	1
O_2	M_1	b_1, b_2, m_1	f_1	252	2
U_2	—	b_2	—	3	1
Max	M_2	b_1, b_2, m_2	r	288	3
Monolithic Synthesis					
O_1, U_1, O_2, U_2, Max	M_0, M_1, M_2	b_1, b_2, m_1, m_2	f, f_1, r	912	6

locations each. The largest compositions are encountered at the end of synthesis for O_1 and Max and have 288 unfolded states each. In contrast, the standard algorithm to construct a supervisor for all plants and specifications together explores a state space of 912 states and produces a single supervisor with six locations.

5 Conclusions

The working paper presents an algorithm that calculates modular controllable supervisors that control a system in a least restrictive way. The proposed incremental approach is simpler and produces smaller supervisors than the usual methods of monolithic synthesis in the literature. It improves on the authors' previous work [13] by completely removing some components and variables from the subsystems subject to synthesis at each step of the algorithm.

In future work, the authors would like to investigate the use of *variable abstraction* [13] to further reduce the subsystems. It would also be interesting to extend the approach to consider the synthesis of nonblocking supervisors.

References

[1] Åkesson, K., Flordal, H., Fabian, M.: Exploiting modularity for synthesis and verification of supervisors. In: Proceedings of the 15th IFAC World Congress on Automatic Control. Barcelona, Spain (2002)

[2] Brandin, B.A., Malik, R., Malik, P.: Incremental verification and synthesis of discrete-event systems guided by counter-examples. IEEE Transactions on Control Systems Technology **12**(3), 387–401 (2004). DOI 10.1109/TCST.2004.824795

[3] Chen, Y., Lin, F.: Modeling of discrete event systems using finite state machines with parameters. In: Proceedings of 2000 IEEE International Conference on Control Applications (CCA), pp. 941–946. Anchorage, Alaska, USA (2000). DOI 10.1109/CCA.2000.897591

[4] Hoare, C.A.R.: Communicating Sequential Processes. Prentice-Hall (1985)

[5] Le Gall, T., Jeannet, B., Marchand, H.: Supervisory control of infinite symbolic systems using abstract interpretation. In: Proceedings of the 46th IEEE Conference on Decision and Control, CDC '05, pp. 30–35 (2005)

[6] Miremadi, S., Åkesson, K., Lennartson, B.: Symbolic computation of reduced guards in supervisory control. IEEE Transactions on Automation Science and Engineering **8**(4), 754–764 (2011). DOI 10.1109/TASE.2011.2146249

[7] Mohajerani, S., Malik, R., Fabian, M.: A framework for compositional nonblocking verification of extended finite-state machines. Discrete Event Dynamic Systems: Theory and Applications (2015). DOI 10.1007/s10626-015-0217-y

[8] Ouedraogo, L., Kumar, R., Malik, R., Åkesson, K.: Nonblocking and safe control of discrete-event systems modeled as extended finite automata. IEEE Transactions on Automation Science and Engineering **8**(3), 560–569 (2011). DOI 10.1109/TASE.2011.2124457

[9] de Queiroz, M.H., Cury, J.E.R.: Modular supervisory control of large scale discrete event systems. In: Proceedings of the 5th International Workshop on Discrete Event Systems, WODES '00, pp. 103–110. Ghent, Belgium (2000)

[10] Ramadge, P.J.G., Wonham, W.M.: The control of discrete event systems. Proceedings of the IEEE **77**(1), 81–98 (1989). DOI 10.1109/5.21072

[11] Shoaei, M.R., Feng, L., Lennartson, B.: Abstractions for nonblocking supervisory control of extended finite automata. In: Proceedings of the 8th International Conference on Automation Science and Engineering, CASE 2012, pp. 364–370. Seoul, South Korea (2012). DOI 10.1109/CoASE.2012.6386446

[12] Tarski, A.: A lattice-theoretical fixpoint theorem and its applications. Pacific Journal of Mathematics **5**(2), 285–309 (1955)

[13] Teixeira, M., Malik, R., Cury, J.E.R., de Queiroz, M.H.: Supervisory control of DES with extended finite-state machines and variable abstraction. IEEE Transactions on Automatic Control **60**(1), 118–129 (2015). DOI 10.1109/TAC.2014.2337411

[14] Wonham, W.M.: Supervisory control of discrete-event systems. Systems Control Group, Department of Electrical Engineering, University of Toronto, ON, Canada; at `http://www.control.utoronto.edu/DES/` (2009)